FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Human-in-the-Loop e Aprendizagem na Negociação Automática: Aplicação num Centro de Controlo Operacional Aéreo

Paula Francisca Ferreira Teixeira

DISSERTAÇÃO

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Eugénio Oliveira

Co-orientadora: Ana Paula Rocha

29 de Julho de 2013

Human-in-the-Loop e Aprendizagem na Negociação Automática: Aplicação num Centro de Controlo Operacional Aéreo

Paula Francisca Ferreira Teixeira

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Henrique Lopes Cardoso

Arguente: Daniel Castro Silva

Vogal: Eugénio Oliveira

---

29 de Julho de 2013

Resumo

A gestão de um dia de operações é uma tarefa complexa para qualquer companhia aérea, embora existam várias fases de planeamento e escalonamento antecedentes que têm por base técnicas de optimização que permitem a elaboração de um plano operacional optimizado. Essa complexidade é justificada pela existência de eventos inesperados perto do dia da operação que, por serem impossíveis de prever durante as fases anteriormente referidas, podem arruinar todo o plano operacional. Nestes casos é necessário encontrar, o mais rapidamente possível, uma solução que minimize todos os custos e atrasos associados. Essa tarefa é denominada Gestão de Rupturas.

O presente trabalho enquadra-se no projecto desenvolvido no Laboratório de Inteligência Artificial e Ciência de Computadores em colaboração com a companhia aérea TAP Portugal, onde foi desenvolvido um sistema multi-agente que visa auxiliar na tarefa de gestão de rupturas. Este sistema tenta encontrar, através da utilização de negociação automática entre agentes, uma solução sub-óptima para o problema inicial do plano operacional. Existem dois tipos de agentes nessa negociação: agentes que apresentam propostas de solução para um problema e um outro agente que avalia essas mesmas propostas e determina qual a melhor solução para o problema específico. A solução vencedora é apresentada a um operador humano.

O propósito deste trabalho é apresentar um processo de aprendizagem que permite aos agentes que apresentam propostas de solução aprender, ao longo de sucessivas rondas, as preferências do agente que os avalia. O processo de aprendizagem desenvolvido está adaptado ao ambiente simultaneamente cooperativo e competitivo onde os agentes se encontram.

É ainda do âmbito deste trabalho desenvolver um método de avaliação das soluções vencedoras. Esta avaliação é fornecida ao sistema pelo operador humano. A esta interacção dá-se o nome de *Human-in-the-Loop*, e o seu propósito é permitir ao operador humano influenciar a função de avaliação de soluções e, por conseguinte, a decisão do agente que efectua tais avaliações.

Uma das consequências do trabalho desenvolvido é a obtenção de propostas mais adequadas simultaneamente às preferências do agente avaliador e às necessidades do operador humano. Esta melhoria traduz-se na qualidade global das soluções a ser aplicadas no contexto real, que minimizam, tanto quanto possível, os custos e atrasos inerentes à alteração do plano inicial.

Os resultados finais deste trabalho foram validados e avaliados por membros do controlo operacional da companhia aérea TAP Portugal. As experiências realizadas permitiram a comparação entre o desempenho das diferentes versões do sistema. Os resultados obtidos permitem afirmar que os objectivos desta dissertação foram atingidos.

Este sistema tem vindo a ser desenvolvido com o apoio da companhia aérea TAP Portugal, a qual disponibilizou os recursos necessários ao desenvolvimento do projecto, nomeadamente dados reais relativos ao plano operacional e rupturas ocorridas.

Abstract

Daily operations management is a hard task for every airline company despite the existence of earlier planning and scheduling phases that allow the elaboration of an optimal operational plan. That complexity lays on the existence of unexpected events which happen close to the day of operation. Once they are impossible to preview during the mentioned planning and scheduling phases, their occurrence may ruin the entire operational plan. In these cases it is necessary to find, as soon as possible, a solution minimizing all the costs and delay associated. This task is named as Disruption Management.

The present work comes in the sequence of the project being developed at Artificial Intelligence and Computer Science Laboratory in cooperation with the TAP Portugal airline company, where a multi-agent system which aims to help at the disruption management task was developed. This system tries to find, through the use of an automated negotiation between agents, a sub-optimal solution for the operational plan's initial problem. There are two types of agents at this negotiation: agents presenting a solution proposal for a problem and another agent responsible for evaluating these same proposals and determining which is the best solution for a specific problem. The winning solution is presented to an human operator.

The purpose of this work is to present a learning process that enables the agents responsible for proposing solutions to learn, along subsequent rounds, the preferences of the agent that is evaluating them. The learning process developed is adapted to the simultaneously cooperative and competitive environment where these agents are.

It is also within the scope of this work to create a method of evaluation of the winning solutions. This evaluation shall be provided to the system by the human operator. To this interaction is given the name Human-in-the-Loop and its purpose is to allow the human operator to influence the evaluation function of the solutions and, therefore, the decision of the agent performing such evaluations.

One of the consequences of the developed work is the achievement of more adequate solutions to both evaluator agent's preferences and to the human operator's necessities. This improvement reflects itself in the global quality of the solutions to be applied in the real context, which minimize, as much as possible, the costs and delays inherent to the change of the initial plan.

The final results of this work were validated and evaluated by members of the operational control center of TAP Portugal airline company. The experiments performed allowed for a comparison between the performance of different versions of the system. The obtained results allow to affirm that the goals of this thesis were accomplished.

This system is being developed with the support of the TAP Portugal airline company, which provided the needed resources to the project development, including real data related to the operational plan and occurred disruptions.

Agradecimentos

Um bem-haja aos meus orientadores, o Professor Eugénio Oliveira e a Professora Ana Paula Rocha, por todo o apoio e colaboração que me prestaram durante o desenvolvimento desta dissertação. Um outro bem-haja ao Engenheiro António Castro, pois sem a sua enorme colaboração e compreensão não teria sido, de todo, possível. Saliento que o Engenheiro António Castro supervisionou esta dissertação como representante da companhia aérea TAP Portugal, à qual devo também expressar o meu agradecimento pela sua colaboração.

Gostaria de agradecer ainda aos colegas do LIACC e da Faculdade que me acompanharam durante esta fase e que, de uma forma ou de outra, me prestaram auxílio. Um agradecimento especial aos colegas Gustavo Laboreiro, Pedro Brandão, Ricardo Teixeira e José Pedro Silva.

Por fim, não poderia deixar de prestar homenagem a toda a minha família, a quem devo muito mais do que apenas a educação.

Francisca Teixeira

Conteúdo

Lista de Figuras

LISTA DE FIGURAS

Lista de Tabelas

Abreviaturas e Símbolos

BC	*Business Class*
CBR	*Case-Based Reasoning*
CTA	Controlo de Tráfego Aéreo
GQN	*Generic Q-Negotiation*
LIACC	Laboratório de Inteligência Artificial e Ciência de Computadores
MASDIMA	*Multi-Agent System for Disruption Management*
NB	*Narrow Body*
WB	*Wide Body*
YC	*Economic Class*

Capítulo 1

Introdução

Este capítulo fornece um enquadramento geral do trabalho que se relata no presente documento. Após uma apresentação do contexto do trabalho, são descritos a motivação, os objectivos e ainda as contribuições científicas consideradas mais relevantes. Por último, é apresentada a estrutura do documento.

1.1 Contexto

Esta dissertação enquadra-se no projecto que tem vindo a ser desenvolvido no Laboratório de Inteligência Artificial e Ciência de Computadores (LIACC), situado na Faculdade de Engenharia da Universidade do Porto, em colaboração com a TAP Portugal.

1.1.1 Gestão de Rupturas

Em qualquer companhia aérea existem várias fases de planeamento cujo objectivo é elaborar um plano e escalonamento óptimo de todos os recursos que a empresa possui, nomeadamente recursos físicos, como aviões, e recursos humanos, como membros da tripulação. Algumas destas fases chegam a acontecer com vários meses de antecedência relativamente ao dia de operações em planeamento, pelo que se torna difícil lidar com qualquer evento inesperado que se verifique perto ou até no próprio dia da operação. Na verdade, a tarefa de controlo de um dia de operações é uma tarefa complexa para os centros de controlo operacional, pois estes eventos inesperados acontecem muito frequentemente e acabam por ter um grande impacto em todo o plano operacional. Não é invulgar que, perto da hora de partida de um dado voo, exista uma anomalia técnica no avião que não se consegue resolver sem atrasar a operação. O mesmo atraso poderá verificar-se quando um membro da tripulação não se apresentar ao trabalho. Em qualquer um dos casos, o plano original já não poderá ser executado, pelo que é necessário pensar numa solução que, tanto quanto possível, minimize os atrasos e custos inerentes ao replaneamento necessário. Sublinha-se que um evento poderá ter um impacto não apenas num voo, mas em vários. Por exemplo, um avião

que parta do aeroporto de origem com um dado atraso, poderá fazer com que os seus passageiros cheguem atrasados a um voo de ligação, pelo que esse voo também sofrerá o impacto do atraso inicial. Logo, é necessário que qualquer problema seja resolvido o mais rapidamente possível. A resolução deste tipo de problemas denomina-se Gestão de Rupturas (*Disruption Management*), sendo que uma ruptura é uma falha no plano operacional da companhia aérea que pode originar atrasos.

A complexidade da gestão de rupturas está na quantidade de variáveis que é necessário considerar para se chegar a uma solução. Qualquer evento inesperado acaba por ter impacto em três perspectivas diferentes que são o avião, a tripulação e os passageiros. Desta forma, a resolução de um problema originado por uma ruptura implica um plano operacional para cada uma destas perspectivas, com a necessidade de se minimizar custos e atrasos. Outro factor que aumenta a complexidade do processo de resolução do problema é o dinamismo do ambiente. Uma solução que possa ser viável num determinado momento, pode já não ser possível de aplicar num momento posterior.

O sistema actualmente em desenvolvimento, denominado MASDIMA[1], pretende dar uma resposta a esta dificuldade dos centros de controlo, utilizando uma abordagem multi-dimensional que visa cobrir as diferentes perspectivas do mesmo problema. Os eventos relativos às múltiplas perspectivas são, portanto, distribuídos e cada perspectiva pode sugerir uma possível solução. Em consequência, considerou-se um processo de negociação automática entre agentes cujo objectivo é encontrar a melhor solução integrada, ou seja, que cubra da melhor forma possível as três perspectivas.

1.1.2 Sistemas Multi-Agente

Um sistema multi-agente, como o próprio nome indica, implica a existência de um ou mais agentes que, podendo ter diferentes responsabilidades e competências, interagem entre si de forma a conseguir alcançar um objectivo comum. Por agente entende-se uma entidade computacional que possui um comportamento próprio, com algum grau de autonomia, dentro de um determinado ambiente, e que por isso difere dos componentes de software que possuem uma interacção pré-definida. Um agente é um elemento que possui algum tipo de conhecimento que o leva a ponderar as consequências de executar uma determinada acção no ambiente envolvente (que pode incluir ou não outros agentes).

O paradigma dos sistemas multi-agente possui importantes características que o distinguem de outros modelos computacionais [Woo09]. Uma das características mais relevantes consiste na autonomia conferida aos agentes: um agente possui controlo sobre o seu comportamento sem que haja interferência directa de um humano ou qualquer outro agente. Por outro lado, o paradigma permite que nenhum dos agentes possua uma visão global do ambiente, pelo que cada uma destas entidades representa apenas uma parte de todo um sistema. Esta característica reduz, desde logo, a complexidade do sistema, tornando-o descentralizado, isto é, nenhum agente tem a possibilidade

[1] Do inglês *Multi-Agent System for Disruption Management*.

de controlar todo o sistema. Outras importantes características são a racionalidade e sociabilidade de um agente, ou seja, se, por um lado, o agente deve comportar-se de forma a maximizar o seu desempenho face a uma função de utilidade própria, por outro deve comunicar com outros agentes e ajudá-los nas suas tarefas. A reactividade é uma característica de sistemas onde existem agentes observadores do ambiente em que estão inseridos e que conseguem reagir atempadamente às mudanças que se vão verificando. A pró-actividade sublinha a importância de um agente reagir não só quando constata alterações no ambiente mas também quando assim lhe for oportuno, pelo que o agente deverá possuir objectivos e tomar a iniciativa de tentar satisfazê-los.

É pela existência das características enunciadas anteriormente que este paradigma se torna adequado no contexto da gestão de rupturas em centros de controlo operacional aéreo.

Através do paradigma de sistemas multi-agente torna-se possível tratar a complexidade de um problema decompondo-o de acordo com várias perspectivas mais simples e atribuir a sua resolução a diferentes entidades que trabalham em paralelo e que, tentando sempre maximizar o seu desempenho, se organizam para resolver o problema da melhor forma possível de um ponto de vista global.

É ainda a pensar no futuro que se adopta este paradigma. Actualmente um problema originado por um ruptura é considerado segundo três perspectivas diferentes. Contudo, pode haver necessidade de considerar o problema adicionando-lhe uma nova perspectiva. Num sistema multi-agente torna-se simples fazer tal alteração, pois o sistema é facilmente escalável. O mesmo acontece no sentido inverso, ou seja, se existir a necessidade de considerar o problema em menos perspectivas, facilmente se retira do sistema o agente responsável por essa perspectiva. A solução proposta também se adequa à possibilidade de algumas das, ou todas as, perspectivas serem tratadas por entidades competitivas que, por isso, desejam manter a privacidade dos cálculos que lhes permite formular e enviar as suas propostas.

1.2 Motivação e Objectivos

O trabalho desta dissertação insere-se no âmbito do desenvolvimento do sistema MASDIMA, o qual se descreve com maior detalhe no capítulo 2. Existe já um protótipo desse sistema que fornece uma boa resposta a um problema no plano operacional em comparação com o tradicional processo de resolução de rupturas nos centros de controlo operacional aéreo. Contudo, após algumas experiências com esse protótipo, foram identificadas certas necessidades de melhoramento que deram origem a esta dissertação.

O enorme dinamismo do ambiente onde o sistema actua, aliado à urgência da necessidade de uma solução, exige que o sistema seja rápido e eficiente na devolução de uma solução para um problema. Para responder a tais requisitos, pretende-se dotar os agentes responsáveis pela procura de soluções de uma capacidade de orientação no espaço de soluções possíveis que lhes permita descartar mais rapidamente aquelas que não são tão vantajosas.

Quanto maior for o grau de automatização de um sistema, maior será o grau de desconfiança que qualquer resposta provocará no meio social em que funciona. Essa desconfiança aumentará

proporcionalmente com o grau de responsabilidade e complexidade do sistema. Tal foi constatado no sistema MASDIMA, o qual não inclui um mecanismo que permita, quer a validação da solução final por um operador humano, quer a sua avaliação. A esta interacção dá-se o nome de *Human-in-the-Loop*, já que exige a interação de um operador humano com o sistema logo após o término do processo de obtenção da solução final. A não aceitação da solução obtida implica a repetição de todo o processo de obtenção de solução. A partir da informação dada pelo operador humano, relativa à avaliação que este faz da solução, o sistema deve ser capaz de rapidamente aprender a encontrar soluções adequadas a tal informação.

Assim, os objectivos desta dissertação podem ser enumerados nos pontos seguintes:

- Aumentar a qualidade das soluções devolvidas sem prejuízo do desempenho do sistema.
- Facilitar a adaptação do sistema MASDIMA ao contexto real.
- Tornar o sistema MASDIMA num sistema socialmente aceite.

1.3 Contribuições Científicas

As contribuições científicas mais relevantes do desenvolvimento desta dissertação são:

- Inclusão de um processo de aprendizagem nos agentes responsáveis pela procura de soluções que participam na negociação automática, permitindo atingir o primeiro dos objectivos especificados. O processo de aprendizagem é adaptado ao ambiente simultaneamente competitivo e cooperativo onde esses agentes se encontram e pode ser utilizado com os algoritmos *Q-Learning* ou *Sarsa* (Capítulo 4).
- Inclusão de um processo que permite a interacção do operador humano com o sistema MASDIMA e a adaptação deste ao contexto real, através da validação e avaliação, por parte do operador humano, das soluções devolvidas pelo sistema, permitindo atingir o segundo e terceiro objectivos especificados (Capítulo 5).

1.4 Estrutura da Dissertação

Para além da presente introdução, esta dissertação contém mais seis capítulos. No capítulo 2 apresenta-se a arquitectura e funcionamento do protótipo do sistema MASDIMA, o que permite identificar os limites dos problemas abordados nesta dissertação. No capítulo 3 é efectuada a revisão do estado da arte, que oferece uma perspectiva sobre os trabalhos desenvolvidos e resultados que se têm vindo a obter na área. Nos capítulos 4 e 5 são descritas as soluções para os dois problemas identificados: aprendizagem no processo de procura de soluções e interacção com um operador humano, respectivamente. No capítulo 6 são apresentadas as experiências realizadas e discutidos os resultados obtidos. Finalmente, no capítulo 7 são apresentadas as conclusões finais relativas a esta dissertação. Nos anexos A, B e C são apresentadas tabelas devidamente referidas ao longo deste documento.

Capítulo 2

Descrição do Problema

Este capítulo tem como função apresentar e delimitar o problema que se propõe resolver nesta dissertação. Para tal, é necessário conhecer a arquitectura e funcionamento do sistema MASDIMA[1], o qual será também apresentado neste capítulo.

2.1 Introdução

O MASDIMA é um sistema de software para a resolução automática de rupturas em planos operacionais aéreos que tem vindo a ser desenvolvido no LIACC[2]. A compreensão da sua arquitectura é um factor determinante para o entendimento do problema que será exposto. O protótipo descrito neste capítulo corresponde à versão funcional existente no início do desenvolvimento desta dissertação. Quando for necessário referi-lo será pelo nome de "Protótipo Base", uma vez que serviu de base ao desenvolvimento desta dissertação. Deve ser tido em conta que a descrição que se segue não inclui todos os detalhes do sistema, incidindo apenas sobre os aspectos fulcrais para o entendimento do problema abordado neste trabalho. Mais detalhes sobre a arquitectura ou sobre qualquer aspecto do funcionamento do MASDIMA estão descritos em [CO11], [CRO12] e [Cas13].

No próximo sub-capítulo apresentar-se-á a arquitectura do sistema MASDIMA. Nos sub-capítulos 2.3 e 2.4 são apresentados os problemas que se pretende resolver. No sub-capítulo 2.5 resumem-se algumas ideias fundamentais.

2.2 Arquitectura do Sistema

O MASDIMA caracteriza-se por ser um sistema multi-agente, ou seja, um sistema onde existem vários agentes inteligentes que interagem entre si. Na figura 2.1 está representada a arqui-

[1] Do inglês *Multi-Agent System for Disruption Management*.
[2] Laboratório de Inteligência Artificial e Ciência de Computadores

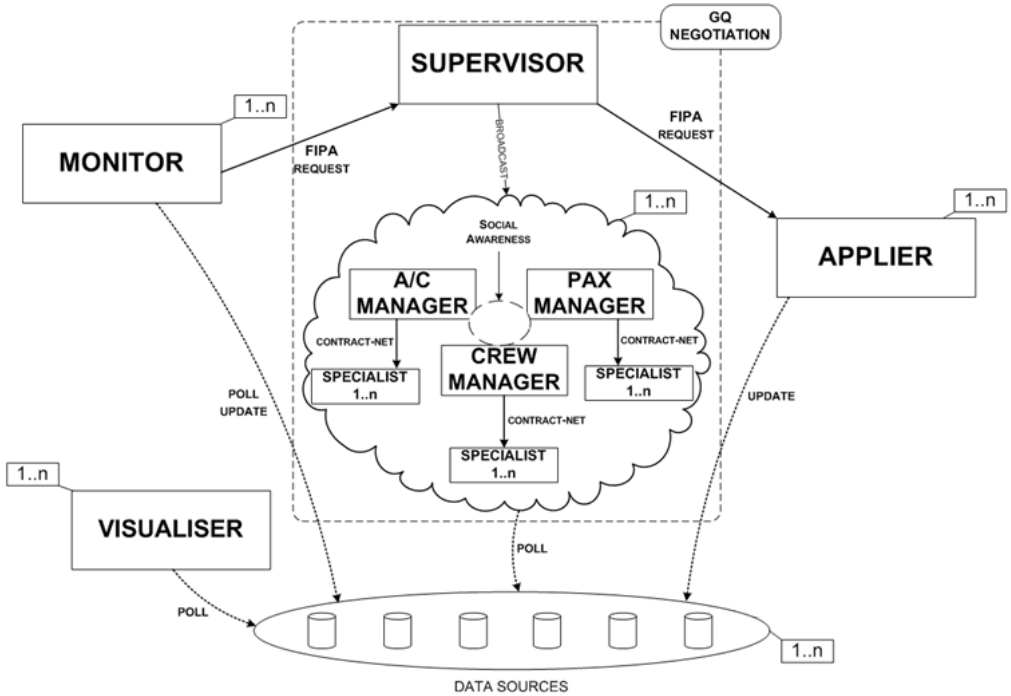

Figura 2.1: Arquitectura do sistema MASDIMA - A arquitectura não sofreu alterações no decurso do desenvolvimento desta dissertação.

tectura do MASDIMA, onde se incluem os agentes *Monitor*, *Supervisor*, *Manager*, *Specialist*, *Visualiser* e *Applier*.

O agente ***Monitor*** está ligado ao plano operacional da companhia aérea, sendo responsável por detectar eventos que possam originar rupturas nesse plano. Quando é detectado um qualquer evento, o agente *Monitor* efectua uma análise que permite detectar as rupturas emergentes e os impactos destas no plano original. As rupturas podem provocar um ou mais atrasos nos voos, caso em que o atraso é considerado um problema. Quando surge um problema o agente *Monitor* deve comunicá-lo ao agente ***Supervisor***. Ao receber a comunicação de um problema, o agente *Supervisor* dá início a um processo de resolução do problema que inclui um protocolo de negociação automática composta por várias rondas, denominado *Generic Q-Negotiation* (GQN). Este protocolo foi desenvolvido pela equipa responsável pelo MASDIMA numa fase anterior ao início desta dissertação e foi pensado para ser um protocolo genérico aplicável também a outros contextos [Cas13]. Os aspectos mais importantes do funcionamento do GQN encontram-se graficamente representados no diagrama da figura 2.2.

Para participar na negociação, o agente *Supervisor* convoca outros três agentes, os quais são conhecidos por agentes *Manager*. Mais concretamente, estes agentes denominam-se ***Aircraft Manager***, ***Crew Member Manager*** e ***Passenger Manager*** por serem responsáveis pela resolução das diferentes perspectivas de um problema: avião (*aircraft*), tripulação (*crew member*) e passageiros (*passenger*), respectivamente. Segundo o protocolo GQN, cada um destes agentes envia uma proposta de solução para o problema que abranja todas as perspectivas, ou seja, uma

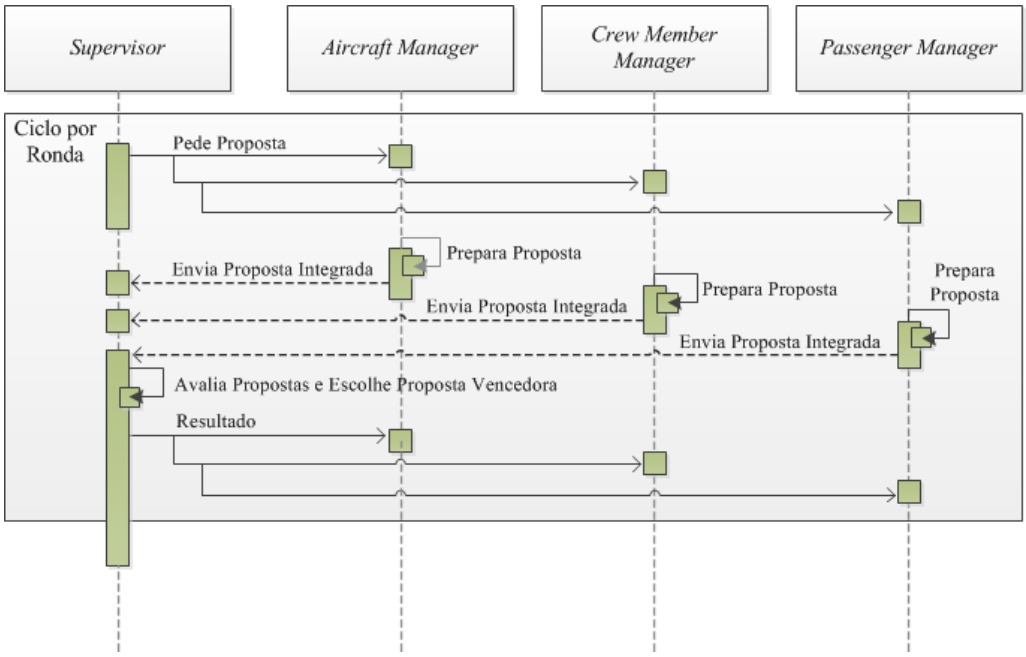

Figura 2.2: Protocolo GQN - Diagrama de sequência que representa apenas a interacção entre o agente *Supervisor* e os agentes *Manager*.

solução integrada. O agente *Supervisor* avalia cada proposta integrada e escolhe, necessariamente, um agente vencedor por ronda. A negociação processa-se em várias rondas, como será descrito ainda neste sub-capítulo.

No entanto, cada agente *Manager* não encontra por si só uma solução que integre as três perspectivas. Ao receber a convocatória, que corresponde ao início de uma ronda da negociação com o agente *Supervisor*, os agentes *Manager* iniciam uma outra negociação entre si. Esta negociação de segundo nível visa integrar numa solução global apenas três soluções parciais, sendo que cada agente *Manager* faz a requisição, aos outros dois agentes, de uma solução parcial que complete a sua segundo as restrições impostas pelo primeiro agente *Manager*.

Os responsáveis pela procura de soluções não são os agentes *Manager*. Cada agente *Manager* possui à sua disposição um ou mais agentes *Specialist*. Um agente *Specialist* apenas possui conhecimento sobre uma perspectiva, a perspectiva do seu agente *Manager*. Cada um dos agentes *Specialist* implementa um dado algoritmo de procura de soluções, pelo que são os verdadeiros responsáveis por procurar soluções parciais. Quando um agente *Manager* recebe uma convocatória da parte do agente *Supervisor*, começa por solicitar aos seus agentes *Specialist* que encontrem um conjunto de soluções parciais para a sua perspectiva.

Cada agente *Specialist* comunica uma lista de soluções possíveis que o agente *Manager* ordena por um valor de utilidade calculado por si, através de uma função de utilidade[3]. Deste conjunto, o agente *Manager* escolhe a solução parcial com maior utilidade e adiciona-lhe uma série de restrições, comunicando-a aos outros dois agentes *Manager* para que estes a completem. Quando estes

[3]Cada agente *Manager* possui a sua própria função de utilidade, que mais nenhum agente conhece, que reflecte, essencialmente, os valores dos custos e atrasos da solução para a sua perspectiva.

dois agentes *Manager* recebem o pedido do primeiro agente *Manager*, tentam procurar soluções para a sua perspectiva que satisfaçam as restrições impostas e comunicam, quando a encontram, a sua parte da solução ao agente *Manager* que fez o pedido. Como todos os agentes *Manager* procedem desta forma, todos eles apresentam uma proposta de uma solução integrada ao agente *Supervisor* para avaliação.

Cada proposta é caracterizada por um plano de domínio[4] para cada perspectiva e por seis atributos, dois por perspectiva, que traduzem os custos e possíveis atrasos associados a cada plano.

$$Sol_{x,i} =< Delay_{ac}, Cost_{ac}, Delay_{cw}, Cost_{cw}, TripTime_{px}, Cost_{px} > \tag{2.1}$$

A solução representada no n-tuplo da expressão 2.1 é um exemplo simplificado da estrutura de uma solução integrada. O agente *Manager* que apresentou a solução $Sol_{x,i}$ é representado pela variável *x*, o número da ronda onde a solução foi apresentada está definido na variável *i*, *ac* diz respeito à perspectiva responsável pelo avião, *cw* à perspectiva da tripulação e *px* representa a perspectiva dos passageiros. O atributo *Delay*, em ambas as perspectivas do avião e dos tripulantes, representa o atraso no voo que a solução implica. No caso da perspectiva que diz respeito aos passageiros, esse atributo não é considerado, havendo, no entanto, o atributo $TripTime$ que representa o atraso no tempo de viagem dos passageiros. O atributo *Cost* representa o custo adicional associado ao plano para a perspectiva a que diz respeito.

Após ter recebido as três diferentes propostas de solução integrada do problema, o *Supervisor* avalia-as segundo uma função de utilidade e valores preferenciais, ambos desconhecidos para os agentes *Manager*. A função de utilidade *U* do agente *Supervisor*, expressa na equação 2.2, é caracterizada por um conjunto de parâmetros associados a cada perspectiva (α) e a cada atributo das diferentes perspectivas (β) que compõe uma solução integrada. Os parâmetros α traduzem a importância dada pelo agente *Supervisor* a cada perspectiva do problema e os parâmetros β traduzem a importância dada aos diferentes atributos das várias perspectivas. O objectivo é maximizar o valor da função U. Os valores preferenciais que o agente *Supervisor* possui são utilizados numa avaliação qualitativa da proposta que é depois comunicada aos agentes *Manager* responsável pela proposta. Esta avaliação qualitativa consiste numa comparação dos valores preferenciais do agentes *Supervisor* com os valores dos atributos da proposta de solução. Os únicos valores preferenciais utilizados na função de utilidade U correspondem ao valores máximos de cada atributo.

A avaliação atribuída a cada proposta, embora seja comunicada a cada agente *Manager*, não revela os valores dos parâmetros da função de utilidade nem os valores preferenciais do agente *Supervisor*. O agente *Manager* saberá apenas se foi o vencedor ou não da ronda e saberá ainda o valor qualitativo atribuído a cada atributo que compõe a proposta. O valor qualitativo poderá ser *Low*, se o valor se apresentar abaixo do valor preferencial do agente *Supervisor*, *Ok* no caso de o valor do parâmetro ser concordante com o valor preferencial, $High$ se o valor do parâmetro for

[4]Um plano de domínio, no caso da perspectiva do avião poderia ser, por exemplo, trocar um avião por outro para um determinado voo. Na perspectiva do tripulante, poderia ser trocar um tripulante pelo outro. No caso da perspectiva do passageiro poderia ser alterar o voo dos passageiros por outro.

elevado e ainda *VeryHigh* para valores que ultrapassem em demasia o valor preferencial.

$$
\begin{aligned}
U =& 1 - \left(\frac{c}{\alpha_{ac} + \alpha_{cw} + \alpha_{px}} \right) \quad U \in [0,1] \\
c =& \alpha_{ac} \left(\frac{\beta_{cost\_ac} \left(\frac{cost\_ac}{max_{cost\_ac}} \right) + \beta_{delay\_ac} \left(\frac{delay\_ac}{max_{delay\_ac}} \right)}{\beta_{cost\_ac} + \beta_{delay\_ac}} \right) \\
&+ \alpha_{cw} \left(\frac{\beta_{cost\_cw} \left(\frac{cost\_cw}{max_{cost\_cw}} \right) + \beta_{delay\_cw} \left(\frac{delay\_cw}{max_{delay\_cw}} \right)}{\beta_{cost\_cw} + \beta_{delay\_cw}} \right) \\
&+ \alpha_{px} \left(\frac{\beta_{cost\_px} \left(\frac{cost\_px}{max_{cost\_px}} \right) + \beta_{tripTime\_px} \left(\frac{tripTime\_px}{max_{tripTime\_px}} \right)}{\beta_{cost\_px} + \beta_{tripTime\_px}} \right)
\end{aligned}
\tag{2.2}
$$

Como já foi referido, o GQN foi pensado para ser aplicável a diferentes contextos. Embora no contexto específico não exista a necessidade de tornar privadas as preferências do agente, optou-se por desenvolver o protocolo dessa forma, pelo que a questão de privacidade de preferências ultrapassa os limites da presente dissertação e constitui uma imposição ao seu desenvolvimento. Mais detalhes sobre esta questão podem ser encontrados no trabalho de Castro [Cas13].

No Protótipo Base os agentes *Manager* já eram dotados de um método que os permitia adaptar a sua nova proposta. Este método tinha em conta a avaliação qualitativa que o agente *Supervisor* atribuía à proposta da ronda anterior. O agente *Manager* que fosse considerado pelo agente *Supervisor* o vencedor de uma dada ronda deveria apresentar, na ronda seguinte, a solução que lhe permitiu ganhar na ronda anterior. No entanto, se o agente *Manager* perdesse a ronda, devia analisar a classificação dos atributos, procurar novas soluções e devolver uma proposta de solução integrada que respeitasse, total ou parcialmente, essa classificação.

$$Aval_{ac,1} = < High, Ok, Ok, Ok, Ok, Ok > \tag{2.3}$$

Como exemplo, considere-se que o agente *Supervisor* atribui a avaliação representada no n-tuplo da expressão 2.3 à solução proposta pelo agente *Aircraft Manager* na primeira ronda. O valor do atributo $Delay_{ac}$ foi classificado como sendo $High$, pelo que o agente deverá, na próxima ronda, propor uma solução em que o atraso desse avião seja menor em relação ao valor da solução anteriormente proposta. Note-se que o agente adapta o seu comportamento momentaneamente, não havendo qualquer aprendizagem com experiências anteriores. Nada coíbe o agente de oscilar entre duas propostas, mesmo perdendo em todas as rondas, se a avaliação atribuída e recebida imediatamente antes estiver de acordo com a próxima solução.

Na primeira ronda de resolução de um problema, os agentes *Manager* não possuem qualquer tipo de avaliação do agente *Supervisor*, pelo que a estratégia de escolha entre as soluções de que dispõem passa por escolher a solução que mais utilidade possui para si. A partir deste momento, os agentes *Manager* receberão do agente *Supervisor* a avaliação que deverão respeitar. No protótipo sem aprendizagem considera-se que uma proposta satisfaz a avaliação atribuída se melhorar pelo

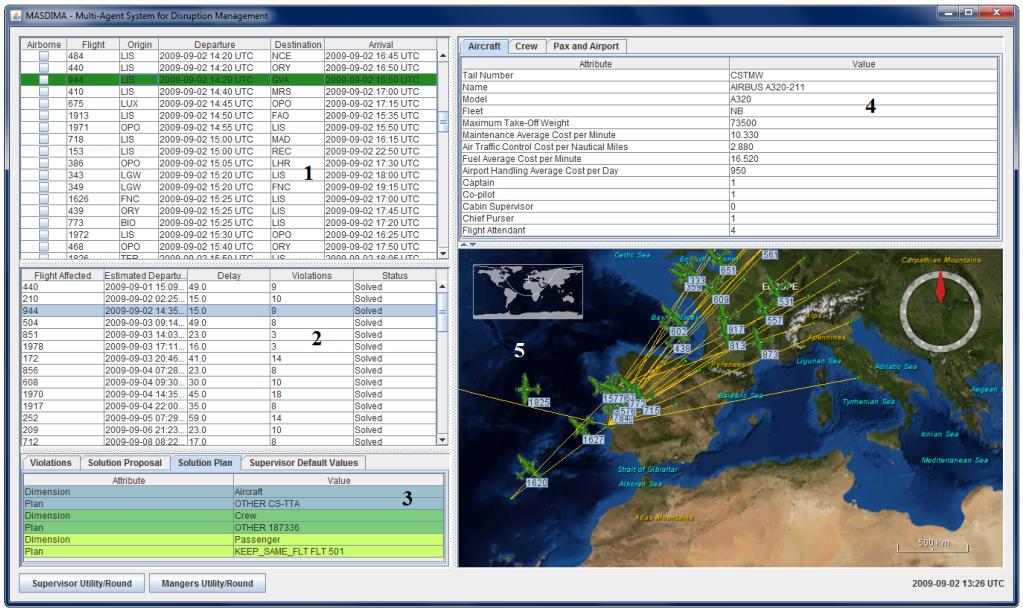

Figura 2.3: Interface do Protótipo Base do MASDIMA - Inclui o plano operacional e informação sobre os recursos de cada voo, os problemas emergentes, as soluções encontradas e ainda a representação gráfica da rota dos voos em curso.

menos um dos atributos segundo a avaliação. Depois de filtrar as soluções recebidas pelo agente *Specialist* em função deste critério, o agente *Manager* volta a ordenar as soluções pela maior utilidade para a sua própria perspectiva e propõe a que maior utilidade possuir. A negociação continua, repetindo-se este processo em múltiplas rondas. Ao fim de um determinado número de rondas, o agente *Supervisor* declara um vencedor, que deverá ser o vencedor da última ronda.

Idealmente, a solução deveria ser enviada para o agente ***Applier***, responsável pela aplicação da solução no plano operacional. Contudo, esta funcionalidade ainda não foi desenvolvida, pelo que a solução é apenas guardada numa base de dados. Terminado este processo, o agente *Supervisor* ficará à espera de uma nova comunicação de um problema pela parte do agente *Monitor*.

Existe ainda um agente ***Visualiser*** que é o responsável por apresentar toda a informação operacional necessária numa interface gráfica, representada na figura 2.3. Nesta são disponibilizadas todas as informações necessárias à monitorização do sistema. Na área *1* são apresentadas informações básicas sobre os voos do plano operacional que se realizam nas próximas 24 horas. A área *4* contém informações sobre os recursos associados a um voo e é actualizada segundo o voo que é seleccionado na área *1*. A área *2* contém a lista de problemas associados ao plano operacional. A área 3 contém informações sobre a solução encontrada pelo sistema para um problema e é actualizada segundo o problema que é seleccionado na área *2*. A área *5* disponibiliza informação gráfica sobre a rota dos voos em curso. Através dos botões dispostos no canto inferior esquerdo, o operador pode aceder a gráficos (representados na figura 2.4) que demonstram o valor de utilidade e o valor dos atributos das várias propostas apresentadas nas rondas de uma negociação.

Figura 2.4: Gráficos de Utilidade das Propostas de Solução - Demonstram a utilidade (individual ou global) das soluções integradas de cada agente *Manager* apresentadas nas diferentes rondas de uma negociação.

2.3 Aprendizagem na Negociação Automática

Antes de avançar na descrição do problema é importante clarificar algumas questões. Neste sub-capítulo apresenta-se com maior detalhe o problema do processo de aprendizagem do lado dos agentes *Manager*. Embora a interacção de um operador humano com o sistema envolva também um processo de aprendizagem, esse tópico será abordado no sub-capítulo 2.4. Esta divisão pretende facilitar a compreensão de ambos os tópicos, bem como a delimitação de cada problema.

Uma das principais características dos sistemas multi-agente é a possibilidade da distribuição do conhecimento pelos agentes existentes. Também no MASDIMA se pode verificar esta característica, pois os agentes *Manager* possuem conhecimento relativo a diferentes perspectivas, por vezes conflituantes, de um problema. Nenhum deles conhece a função de utilidade ou valores preferenciais dos outros agentes *Manager* nem do agente *Supervisor*. Esta característica permite classificar os agentes que geram as propostas segundo as diferentes perspectivas em agentes competitivos. O objectivo de cada um destes agentes é ser considerado o vencedor de cada ronda, pelo agente Supervisor, através da apresentação de uma solução integrada em que a utilidade da sua perspectiva seja assegurada. Neste sentido, cada agente deverá tentar apresentar propostas que possuam a mais elevada classificação por parte do agente *Supervisor*. Como não conhecem qualquer preferência do *Supervisor* nem sabem qual a função de utilidade que este usa, a única informação que lhes permite avançar na direcção dessas preferências é o facto de vencerem ou não a ronda, aliado à classificação qualitativa dada a cada um dos atributos das suas propostas de solu-

ção. É através de uma análise à informação, dada iterativamente por cada episódio de negociação (ou seja, por cada ronda), que os agentes *Manager* poderão aprender as preferências do agente *Supervisor* e adaptar as propostas que lhe enviam.

No entanto, embora os agentes *Manager* compitam entre si de forma a ganhar a negociação, há que ter em conta que é a cooperação que entre eles existe que lhes permite apresentar uma solução integrada, pelo que aquilo que um agente ganha ou perde não depende só do que este aprende, mas daquilo que todos os agentes *Manager* aprendem. Deste facto emergem importantes questões que devem ser consideradas aquando da conceptualização de um processo de aprendizagem:

- Qual o processo de aprendizagem mais adequado ao contexto apresentado, sabendo que os agentes cooperam entre si ao mesmo tempo que competem?
- Deverão existir diferentes processos de aprendizagem concorrentes ou será possível implementar um processo centralizado para os vários agentes *Manager*?
- Quais as implicações inerentes a cada tipo de aprendizagem?
- Uma vez que todos os agentes aprendem simultaneamente, qual será o risco de tornar inválidos os processos de aprendizagem?

A resposta às três primeiras perguntas pode ser encontrada no capítulo 3, onde a revisão do estado da arte permitiu projectar uma solução para o problema. A resposta à última pergunta só foi possível após a implementação da solução. Assim, no capítulo 4 podem ser encontradas as conclusões relativamente a essa questão.

2.4 *Human-in-the-Loop*

No Protótipo Base do sistema MASDIMA, no fim de cada negociação, o agente *Supervisor* recolhe a melhor solução e guarda-a numa base de dados. Conceptualmente, essa solução seria aplicada na realidade, resolvendo assim a ruptura no plano operacional. No entanto, a interacção humana existente neste processo de procura e escolha de uma solução é nula, o que causa desconfiança no meio social em que o sistema deverá funcionar. Depois de algumas entrevistas com os membros do centro de controlo operacional aéreo da TAP Portugal, foi identificada a necessidade de um processo que permita a validação da solução devolvida pelo sistema por parte de um operador humano. Essa validação pode corresponder à aceitação ou à rejeição da solução. Neste último caso, o processo deve permitir indicar informação que permita ao sistema ajustar o processo de procura de soluções às necessidades que se verificam no ambiente real.

Como resposta a este requisito o sistema deve apresentar, no final de cada negociação, a solução escolhida ao operador humano. Torna-se agora necessário perceber que informações deve o sistema apresentar ao operador e que informação deverá este fornecer ao sistema, quer a solução seja aceite ou não. Sendo a solução aceite, esta deverá ser guardada na base de dados e posteriormente aplicada no contexto real. É no caso de a solução não ser aceite que se prevêem

as maiores alterações. Será necessário um método que permita ao agente *Supervisor* modificar os parâmetros da sua própria função de utilidade, os seus valores preferenciais, ou até ambos, dependendo sempre do método de classificação que se implementar na parte de interacção com o operador humano. O agente *Supervisor* iniciará novamente a negociação para a resolução do mesmo problema, efectuando agora a avaliação das propostas com a nova função de utilidade (com os parâmetros alterados).

Existe, não obstante, alguma controvérsia em relação à inclusão de um humano num sistema cujo objectivo é exactamente colmatar a dificuldade que o humano sente no desempenho de determinada tarefa. No caso do MASDIMA, a ideia por trás de todo o sistema é proporcionar uma análise sobre todas as restrições que, dado o elevado número, não são tão eficientemente ponderadas por um operador humano. Logo, a inclusão de uma avaliação por parte de um operador humano pode influenciar o sistema num sentido negativo, o que se pretende evitar.

Surgem assim as seguintes dúvidas:

- Qual deverá ser o impacto no sistema da contribuição do operador humano?
- Que informação deve o operador humano comunicar ao agente *Supervisor*?
- Qual é o efeito da alteração da forma de avaliação do agente *Supervisor* nos processos de aprendizagem que se pretende implementar nos agentes *Manager*?

A revisão do estado da arte permitiu perceber qual a solução que deveria ser aplicada para satisfazer estas exigências. Assim, é ainda no capítulo 3 que se apresentam algumas respostas a estas perguntas. Todavia, e à semelhança da aprendizagem na negociação automática, algumas das questões só puderam ser respondidas após a implementação da solução prevista, pelo que é no capítulo 5 que descrevemos essas conclusões.

2.5 Resumo

Neste capítulo foi apresentada a arquitectura do sistema MASDIMA, o que permitiu contextualizar e melhor definir o problema que se propõe resolver.

Discutiu-se a necessidade de implementar um processo de aprendizagem do lado dos agentes *Manager*, os quais são responsáveis pela apresentação de propostas ao agente *Supervisor*. Concluiu-se que o processo deve permitir que os agentes *Manager* consigam orientar-se num caminho mais promissor dentro do espaço de soluções possíveis, descartando soluções que à partida não serão vantajosas para o sistema e melhorando, assim, a qualidade das propostas efectuadas. Contudo, o facto de os agentes *Manager* se encontrarem num ambiente simultaneamente cooperativo e competitivo levanta algumas questões que é necessário discutir, o que será feito no próximo capítulo.

Foi identificada ainda a necessidade de interacção de um operador humano com o sistema. Definiu-se o objectivo desta funcionalidade em duas facetas. Por um lado, a funcionalidade deve tornar o MASDIMA num sistema socialmente aceite no contexto onde se pretende integrá-lo. Por

outro, pretende-se tornar todo o sistema mais flexível às necessidades do contexto real de um centro de controlo operacional aéreo. Assim, é necessário permitir ao operador validar as soluções encontradas e, no caso de não as aceitar, fornecer informação apropriada que permita ao sistema adaptar-se às exigências do operador humano.

Capítulo 3

Estado da Arte

Neste capítulo apresenta-se a revisão do estado da arte nas áreas delimitadas pela aprendizagem em sistemas multi-agente e pela interacção de um operador humano com um sistema automatizado. É aqui feita uma descrição e comparação de alguns dos trabalhos que têm vindo a ser desenvolvidos na área de sistemas multi-agente que se mostraram mais relacionados com o propósito da presente dissertação.

3.1 Introdução

A área cientifica e técnica que engloba os estudos sobre todas as fases relacionadas com o planeamento e escalonamento de recursos de uma companhia aérea está amplamente estudada e podem ser encontrados inúmeros trabalhos sobre os mais variados tópicos [Pin12]. Contudo, a gestão de rupturas continua a ser um campo pouco aprofundado, devido talvez à complexidade que representa a obtenção de uma solução integrada em tempo útil e de qualidade. Das abordagens conhecidas são poucas as que consideram mais do que uma das três perspectivas em que o problema pode ser decomposto (já definidas anteriormente: avião, tripulação e passageiros) [CO11]. Mesmo as abordagens que tentam resolver o problema para duas das perspectivas utilizam um método de decisão sequencial onde primeiro se resolve o problema para uma dada perspectiva e depois se fornece o resultado dessa primeira fase como problema inicial na seguinte. Assim, a perspectiva que providenciar uma solução em primeiro lugar ficará sujeita a muito menos restrições que as seguintes, pelo que não existe uma equidade na importância dada a cada perspectiva. Ambas as abordagens referidas em [PSJ+10] e [ESB10] são um exemplo desta ordem de resolução naturalmente imposta.

O MASDIMA[1] contorna esta situação através do processo de negociação entre os agentes responsáveis por cada uma das perspectivas. Este processo permite que as diferentes perspectivas tenham, à partida, uma equidade na importância e consideração de restrições. Outra vantagem é

[1] Do inglês *Multi-Agent System for Disruption Management*.

o aumento do conjunto de soluções possíveis através da apresentação de soluções que não podem ser elaboradas num processo de decisão sequencial.

Pelas razões indicadas, o MASDIMA é já um sistema inovador. Da literatura revista na área de gestão de rupturas não existe um sistema que se possa comparar com o sistema MASDIMA em toda a sua extensão. Para uma boa comparação entre as várias abordagens conhecidas sugere-se a leitura de [CLLR10].

Contudo, o objectivo fundamental deste trabalho não é a abordagem utilizada na gestão das rupturas, mas sim a inclusão neste cenário de aprendizagem num contexto de uma negociação automática bem como a interacção de um humano com o sistema. Assim, o estado da arte que será apresentado incidirá essencialmente sobre duas questões: aprendizagem em sistemas multi-agente e interacção do humano com um sistema automático.

Quanto à questão da aprendizagem, importa relembrar que, tal como foi referido no capítulo 2, existem duas situações distintas de aplicação do processo de aprendizagem neste trabalho. Numa delas, a aprendizagem será implementada do lado dos agentes *Manager*, a partir da classificação atribuída às propostas enviadas por estes agentes agora devolvida pelo agente *Supervisor*. A outra situação diz respeito ao agente *Supervisor*, cujo processo de aprendizagem será influenciado pela avaliação extraída da interacção do sistema com o operador humano. Assim, este último processo de aprendizagem encontra-se intrinsecamente associado ao tópico da interacção do sistema com o operador humano.

O resto deste capítulo está estruturado da seguinte forma: no sub-capítulo 3.2 discute-se a aplicação de aprendizagem em sistemas multi-agente e apresentam-se alguns trabalhos desenvolvidos nessa área. No sub-capítulo 3.3 discute-se o tópico sobre interacção de um humano com um sistema e referem-se alguns trabalhos que serviram de base ao desenvolvimento desta dissertação. Por fim, no sub-capítulo 3.4 encontram-se resumidas as ideias base deste capítulo.

3.2 Aprendizagem

O processo de aprendizagem em sistemas multi-agente tem sido realizado seguindo diferentes métodos: aprendizagem supervisionada, aprendizagem não supervisionada e aprendizagem por reforço [RN09] [SB98].

A aprendizagem supervisionada contempla a existência de um supervisor que indica ao agente aprendiz qual deve ser o resultado obtido, pelo que a aprendizagem é feita através de uma comparação da informação obtida pelo sistema com a informação apresentada pelo supervisor. Este tipo de aprendizagem parece não se adequar ao MASDIMA, pois o ambiente em questão é demasiado dinâmico e poderia levar à ineficácia de obtenção de uma solução. Sutton & Barto [SB98] afirmam que esta categoria de aprendizagem não é a mais indicada para aprendizagem através da interacção entre agentes. Ainda assim, existem alguns exemplos notáveis de sistemas multi-agente onde existe uma supervisão mútua entre agentes como, por exemplo, em [GR95].

Na aprendizagem não supervisionada, após a execução de uma acção, o agente analisa o ambiente após tal execução e retira as suas próprias conclusões. Este tipo de aprendizagem é mais

adequada ao pré-processamento de grandes quantidades de dados pouco estruturados e está intrinsecamente ligada a técnicas de *data-mining*. Actualmente, está em desenvolvimento uma solução que utiliza CBR[2] no MASDIMA. Essa aprendizagem dá-se do lado do agente responsável pela avaliação das propostas de solução (o agente *Supervisor*), que possui o histórico das negociações anteriores, e diferencia-se assim do objectivo deste trabalho. Os detalhes desta implementação poderão ser consultados em [Sil13].

Na aprendizagem por reforço é calculada a utilidade de se executar uma qualquer acção a partir de um dado estado, pelo que o agente aprende através da experiência qual a melhor a acção a realizar quando se encontra em determinado estado [SB98]. Ao estar num dado estado e e seleccionando uma acção a, que é executada, o agente chega a um novo estado e'. Dependendo do estado a que chegou receberá uma recompensa r, que poderá valorizar ou penalizar a acção executada.

Tendo em conta as situações em que se pretende usar aprendizagem no presente trabalho, é este tipo de aprendizagem aquele que mais se adequa ao contexto aqui apresentado. Por esta razão, descrever-se-á mais detalhadamente a aprendizagem por reforço no sub-capítulo 3.2.1.

O estudo desenvolvido por Rocha [Roc01] descreve um sistema multi-agente cujos agentes representam empresas independentes com características apropriadas ao contexto de uma actividade comercial num ambiente competitivo e que deverão negociar entre si. O protocolo de negociação proposto prevê uma arquitectura de um para muitos e suporta a execução de várias rondas. A autora refere-se à aprendizagem como uma característica essencial a incluir na negociação num mercado electrónico para que as empresas participantes se consigam adaptar às constantes alterações do mercado. A aprendizagem é feita através da utilização do algoritmo *Q-Learning* [WD92], que será descrito no sub-capítulo 3.2.1, e é por essa razão que o protocolo se denomina Negociação-Q. Neste protocolo, os agentes aprendem a melhorar a sua proposta respeitando, tanto quanto possível, uma avaliação qualitativa que lhes é enviada em resposta à proposta efectuada. Essa avaliação, aliada ao facto de terem sido vencedores ou perdedores da ronda permite o cálculo de uma recompensa que penalizará a execução de más acções, ou seja, a apresentação de más proposta. Após a implementação de aprendizagem por reforço através de uma adaptação ao algoritmo *Q-Learning*, os resultados obtidos levaram a considerar a capacidade de aprendizagem como uma característica relevante para os agentes empresa.

Oliver [Oli96] descreve um outro sistema orientado ao comércio electrónico baseado em negociação bilateral por rondas onde utiliza algoritmos genéticos [Gol89] para a formação de novas estratégias de negociação, sendo utilizado o método de aprendizagem por reforço. O autor classifica os resultados obtidos como promissores, especialmente no âmbito do comércio electrónico, embora seja sublinhado que a simplicidade da negociação estabelecida possa não abranger a complexidade das estratégias praticadas em negociações entre humanos.

Takadama & Fujita [TF04] apresentam algumas conclusões sobre a aplicação de um processo de aprendizagem num sistema de mútua negociação composto por dois agentes. Nas várias experiências desenvolvidas foram aplicados alternadamente dois algoritmos de aprendizagem por

[2]Do inglês Case-Based Reasoning: método de aprendizagem não supervisionada.

reforço: *Q-Learning* e *Sarsa* [SB98]. Os autores sublinham que, embora os prós e contras dos dois algoritmos sejam muito semelhantes, os agentes desenvolveram um comportamento racional aquando da utilização do algoritmo *Q-Learning*, algo que não foi sentido na utilização do algoritmo *Sarsa*.

Embora o contexto seja diferente do aqui apresentado, os autores de [SSK05] apresentam um processo de aprendizagem por reforço aplicado a um ambiente que simula um jogo de futebol. O objectivo da equipa *"keepers"* é controlar e manter a bola entre os seus membros o maior tempo possível, enquanto que a equipa *"takers"* tenta retirar-lhe a bola. Neste sistema, os agentes da equipa *"keepers"* são obrigados, não só a competir com os agentes da equipa adversária, como a cooperar com os agentes da sua própria equipa, pelo que aprendem individualmente as situações em que devem manter ou passar bola. Os autores referem que este estudo apresenta imensos desafios à categoria de aprendizagem por reforço, enumerando a elevada dimensão do espaço de estados, a noção de estado incerto e escondido, a existência de agentes independentes a aprender simultaneamente e ainda a existência de grandes e muito variáveis atrasos na obtenção de resultados das acções tomadas. É usado o algoritmo *Sarsa*(λ) com uma função de *discretização* de espaços que utiliza a técnica *tile-coding* para aprender decisões de alto nível [SS05]. Os autores dizem ter conseguido um óptimo desempenho após pouco treino, embora não existam evidências teóricas que garantam o sucesso do algoritmo no domínio em questão.

3.2.1 Aprendizagem por Reforço

A aprendizagem por reforço é implementada através dos conceitos de estado, acção e recompensa. Um agente, ao encontrar-se num determinado estado, pode optar por executar várias acções no ambiente em que está integrado. Seleccionando uma dessas acções, executa-a. Verifica o estado a que chegou através da comunicação por parte do ambiente, sendo-lhe atribuída uma recompensa cujo valor varia em função da qualidade do novo estado.

Dentro da aprendizagem por reforço, Kaelbling et al [KLM96] identificam duas estratégias para a resolução de problemas. A primeira passa por procurar no espaço de acções aquelas que possuem um melhor desempenho no ambiente e está relacionada com computação evolucionária.

A segunda estratégia utiliza técnicas estatísticas e programação dinâmica para estimar a utilidade de se tomar uma dada acção quando se está num determinado estado. Sutton et al [SB98] agrupam os algoritmos que permitem esta estratégia em três diferentes classes: algoritmos de programação dinâmica, métodos de Monte Carlo e algoritmos de diferença temporal (DT). Segundo estes autores, os algoritmos dinâmicos, embora muito bem desenvolvidos de um ponto de vista matemático, requerem a existência de um modelo preciso do ambiente e são computacionalmente muito elevados. Já os métodos de Monte Carlo, embora não necessitem de um modelo do ambiente, não são adequados a computação incremental, pois aprendem apenas ao fim de cada episódio. Os algoritmos de diferença temporal são o resultado da mistura de ideias de programação dinâmica com as ideias dos métodos de Monte Carlo. Estas são as duas grandes vantagens dos algoritmos de diferença temporal sobre as duas outras categorias, e que apontam para a sua utilização na implementação de aprendizagem no sistema aqui apresentado.

De entre os vários algoritmos da categoria de aprendizagem por reforço que usam diferença temporal destacam-se o *Q-Learning* e o *Sarsa*. Em ambos os algoritmos existem valores Q associados a cada par estado-acção que representam a utilidade de executar uma dada acção quando se está num determinado estado. Segundo Sutton & Barto [SB98], a convergência de ambos os algoritmos para a solução óptima é garantida se todos os estados que compõem o espaço de estados forem visitados um número suficiente de vezes. A diferença entre os dois consiste na forma de selecção da acção a executar e no cálculo da actualização do valor Q do par estado-acção seleccionado em determinado instante.

$$Q-Learning: Q(e,a) = Q(e,a) + \alpha[R + \gamma \max_{a' \in A(e')} Q(e',a') - Q(e,a)] \tag{3.1}$$

$$Sarsa: Q(e,a) = Q(e,a) + \alpha[R + \gamma Q(e',a') - Q(e,a)] \tag{3.2}$$

As equações 3.1 e 3.2 representam a forma de actualização do valor Q de ambos os algoritmos. $Q(e,a)$ representa o valor da qualidade do par composto pelo estado e e acção a. O parâmetro α varia entre 0 e 1 e representa a taxa de aprendizagem do algoritmo, pelo que quanto maior for o seu valor, maior será a contribuição dada pela execução dessa acção a nesse estado e na actualização do valor Q respectivo. O parâmetro γ é uma constante cujo valor se encontra no intervalo $[0,1]$. Representa o valor relativo de recompensa futuras, ou seja, recompensas futuras têm menor peso que recompensas actuais quando este parâmetro é incluído. R representa a função de recompensa, que terá de ser definida de acordo com a forma de avaliação desejada. e' representa o estado a que se chegou a partir da execução da acção a no estado e'.

A diferença entre estas equações reside na parcela que corresponde ao valor Q do próximo par estado-acção, que é composto pelo estado e' e pela acção a'. No *Q-Learning*, é utilizada uma estratégia gananciosa: o par estado-acção que é utilizado nesta parcela corresponde ao par com o maior valor que se pode obter no estado e' a que agora se chegou. Não quer dizer, no entanto, que se seleccione a acção correspondente a esse par na próxima iteração. Este algoritmo actualiza os valores Q baseado em acções hipotéticas, que talvez nunca tenham sido realmente experimentadas. Em contraste, o algoritmo *Sarsa* actualiza o valor Q dos seus pares estado-acção através da experiência, já que a acção a' utilizada na equação 3.2 será realmente executada, e o seu valor actualizado. Mais, as acções futuras são escolhidas de acordo com a mesma política utilizada para seleccionar a acção anterior, algo que não acontece no algoritmo *Q-Learning*.

Pelas razões anteriormente mencionadas, diz-se que o algoritmo *Sarsa* é do tipo *on-policy*, enquanto que o algoritmo *Q-Learning* é do tipo *off-policy*.

A estratégia de selecção de uma acção é uma das principais características da aprendizagem por reforço. Se, por um lado, o agente deve seleccionar os estados mais promissores (observação), por outro, deve também optar por uma atitude exploratória que lhe permita descobrir novos estados que talvez produzam melhor resultados. Das várias estratégias existentes para a selecção da acção a executar, seleccionaram-se as estratégias $\varepsilon - greedy$ e *Softmax* para a implementação de ambos os algoritmos.

A estratégia $\varepsilon - greedy$ deve o seu nome à utilização da constante ε que tem um valor no intervalo [0, 1]. É uma estratégia gananciosa que selecciona, em $1 - \varepsilon$ das vezes, a acção com maior valor Q, sendo que é seleccionada uma outra acção aleatoriamente, com uma distribuição uniforme, nas restantes ε vezes. É esta última característica que lhe confere um traço exploratório.

A estratégia aleatória *Softmax* é bastante semelhante à estratégia $\varepsilon - greedy$. No entanto, em vez de distribuir uniformemente a probabilidade entre as acções a escolher, calcula essa probabilidade em função do valor estimado da acção. Desta forma, acções que possuam maior valor estimado terão maior probabilidade de ser seleccionadas. A distribuição da probabilidade, nesta estratégia, é feita segundo a distribuição de Boltzmann [KLM96], cuja fórmula está representada em 3.3.

$$p(a) = \frac{e^{Q(e,a)/t}}{\sum_{b \varepsilon A} e^{Q(e,b)/t}}, t > 0 \tag{3.3}$$

$p(a)$ representa a probabilidade da acção a ser seleccionada a partir do estado e. A representa o conjunto de acções possíveis a partir do estado e. Nesta distribuição, a exploração é controlada pelo parâmetro t, o qual é denominado de temperatura. Este parâmetro possui um valore inicial elevado que vai decrescendo com o tempo, à medida que o número de iterações cresce. Quando o valor deste parâmetro é elevado, a probabilidade de uma acção ser seleccionada tende a ser distribuída de uma forma mais uniforme por todas as acções. À medida que o valor de t decresce, a probabilidade de uma acção ser seleccionada vai sendo distribuída pelo conjunto de acções em função do valor Q de cada acção, sendo que as acções com maior valor Q passam a ter mais probabilidade de serem seleccionadas.

Ambas as distribuições parecem funcionar bem com qualquer um dos algoritmos seleccionados [SB98], pelo que a escolha entre a distribuição a utilizar dependerá dos resultados das experiências realizadas.

3.2.2 Aprendizagem Concorrente e Aprendizagem por Equipa

Os sistemas referidos no início deste capítulo têm uma importante característica em comum. Todos eles implementam o processo de aprendizagem em mais do que um agente. Os agentes estão, por isso, a aprender em simultâneo. Existe uma clara diferença entre este tipo de sistemas e os sistemas multi-agente onde apenas um dos agentes está a aprender, enquanto que os outros agentes mantêm a sua estratégia fixa. Nesta situação, o agente que aprende não tem que se adaptar a novas estratégias. No caso de haver mais do que um agente a aprender, a aprendizagem de um agente pode acabar por afectar a aprendizagem de um segundo agente e vice-versa. Este é precisamente o caso do sistema MASDIMA, onde os agentes *Manager* devem aprender simultaneamente.

Panait & Luke [PL05] elaboraram um estudo sobre aprendizagem em sistemas multi-agente onde focam principalmente agentes cooperativos, mas referem também o impacto da aprendizagem em sistemas compostos por agentes competitivos. Um sistema multi-agente cooperativo é definido como sendo um sistema onde vários agentes tentam, através da interacção, resolver tarefas em conjunto ou maximizar uma utilidade global. Nestes sistemas, os autores consideram o uso de aprendizagem em duas categorias: aprendizagem por equipa e aprendizagem concorrente. Na

aprendizagem por equipa existe apenas um processo de aprendizagem que determina o comportamento de todos os agentes que pertencem a essa mesma equipa. Já na aprendizagem concorrente, cada agente possui o seu próprio processo de aprendizagem que vai decorrendo em paralelo aos dos outros agentes. A principal vantagem da abordagem concorrente sobre a aprendizagem por equipa, dizem os autores, é a redução do espaço de estados através da separação dos estados de cada agente.

Jasen & Wigand [JW03] referem que o uso de aprendizagem concorrente pode ser preferível em domínios onde a decomposição é possível e vantajosa e quando é útil que cada agente se foque num dado sub-problema com um certo grau de independência dos outros agentes. No entanto, a principal característica desta abordagem, e que surge como o principal desafio, dizem ainda os autores Panait & Luke [PL05], é o facto de cada agente adaptar o seu comportamento no contexto de outros agentes adaptativos e sobre os quais não possui qualquer controlo. No decorrer do processo de aprendizagem, os agentes vão modificando o seu comportamento e isso pode, por sua vez, arruinar o processo de aprendizagem de outros agentes ao tornar obsoletas todas as premissas em que este é baseado.

Na opinião de Panait & Luke [PL05], existem três temas importantes a considerar quando se utiliza a aprendizagem concorrente: atribuição de recompensas, dinâmica de aprendizagem e modelação de outros agentes.

A atribuição de recompensas aborda a forma de distribuição das recompensas individuais pelos agentes quando estes resolvem tarefas em equipa. Balch [Bal98] elaborou um estudo sobre o impacto de se atribuir recompensas locais ou globais a uma equipa de agentes que cooperam entre si em vários domínios. Uma função de recompensa global, diz o autor, é aquela que atribui a todos os membros de uma equipa o mesmo valor de recompensa, enquanto que uma função de recompensa local atribui um valor a cada agente de acordo com o seu desempenho individual. Após realizadas várias experiências, o autor conclui que, embora uma função de recompensa local permita uma aprendizagem mais rápida, os resultados obtidos não são necessariamente melhores do que aqueles obtidos através de uma recompensa global, sublinhando até o aparecimento de estratégias gananciosas que aumentam as recompensas individuais mas baixam o desempenho global do sistema.

A dinâmica de aprendizagem tenta estudar o impacto da co-adaptação em processos de aprendizagem. No que toca à aprendizagem concorrente em ambientes cooperativos, que é frequentemente analisada a partir de uma perspectiva da teoria de jogos, verifica-se que os agentes acabam por desenvolver uma estratégia que os leva a convergir para um equilíbrio de Nash[3], não havendo recompensa suficientemente incentivante que os leve a alterar essa estratégia. Este equilíbrio tem os seus benefícios, pois permite que nenhum dos agentes tenha uma recompensa demasiado baixa. Um bom exemplo desta situação é o conhecido dilema do prisioneiro. No entanto, esse equilíbrio pode corresponder a um comportamento sub-óptimo da equipa como um todo. Os autores

[3] O equilíbrio de Nash [VNM07] é atingido quando nenhum dos agentes pode obter qualquer vantagem alterando o seu comportamento unilateralmente, isto é, dada a estratégia fixa dos outros, o agente nada lucrará com a alteração da sua própria estratégia.

Panait & Luke [PL05] referem que não é fácil criar estratégias que permitam evitar esta convergência. Relativamente a ambientes competitivos que utilizam aprendizagem concorrente, é referida a existência frequente de um agente dominante. Nestas situações, por mais informação que se comunique, nenhum agente consegue evoluir. O agente que ganhou uma vez acaba por ganhar sempre, bem como os agentes que perdem e que permanecem perdedores, independentemente do que possam aprender. Outro fenómeno relatado é o desenvolvimento de uma estratégia que converge para um comportamento cíclico por parte dos agentes.

Nunes & Oliveira, em [NO05] e [NO04], descrevem ainda uma outra forma de aprendizagem cooperativa entre agentes. Essa aprendizagem baseia-se na troca de conselhos entre agentes responsáveis pela resolução de problemas semelhantes. Os agentes possuem duas formas de aprendizagem. Por um lado, existe um processo de aprendizagem por reforço que leva um determinado agente a executar uma acção quando está num determinado estado e a receber a recompensa que lhe é atribuída pela execução dessa acção no ambiente. Por outro lado, o agente poderá apresentar esse mesmo estado a um outro agente que possui melhor desempenho num problema semelhante e que será, por isso, o seu conselheiro. O conselheiro apresentar-lhe-á uma acção que será vista pelo agente que pediu o conselho como a resposta correcta para o problema. Este processo consiste assim num processo de aprendizagem supervisionada, onde o agente conselheiro age como supervisor.

O tópico de modelação dos agentes de uma equipa, abordado por Panait & Luke [PL05], visa dotar um agente de conhecimento suficiente para que este consiga prever as acções que os outros agentes irão executar. Segundo Suryadi & Gmytrasiewicz [SG99], uma das principais motivações de pesquisas neste âmbito é a procura de técnicas que permitam coordenação entre agentes de forma a que as suas acções racionais individuais não tenham efeitos adversos na eficiência global do sistema. Esta é descrita como uma importante aérea a considerar aquando da utilização da abordagem de aprendizagem concorrente. A abordagem é utilizada pelos autores Boutilier & Chalkiadakis [Bou96] [CB03], que empregam um método de aprendizagem *Bayesiano* nos agentes para que estes consigam estimar o actual comportamento ou situação dos seus colaboradores. Desta forma, o agente poderá moldar-se à situação dos agentes colaboradores e tentar cooperar melhor com estes. Considera-se que esta é uma importante abordagem a explorar e que poderá ter um efeito positivo no desempenho dos agentes que compõem o sistema.

3.3 *Human-in-the-Loop*

Nos últimos anos tem-se apostado na automatização de processos como forma de minimizar a introdução de erro humano num processo complexo. Embora não existam dúvidas quanto aos benefícios da automatização nesse sentido, a experiência veio a comprovar que um processo automatizado não substitui de forma absoluta a actividade de um operador humano. O que faz é alterar a forma de contribuição deste em todo o processo [PSW00].

Sheridan & Verplank [SV78] propuseram uma escala que classifica um dado sistema segundo o nível de automatização que este emprega. Essa escala, apresentada na tabela 3.1, é composta

Tabela 3.1: Níveis de Automatização de Sistemas - Níveis de automatização de decisão, selecção e execução de uma acção num sistema.

Nível	Descrição
10	O sistema decide tudo e age autonomamente, ignorando o humano.
9	O sistema informa o humano se assim o decidir.
8	O sistema apenas informa o humano se este lho pedir.
7	O sistema executa automaticamente e informa posteriormente o humano.
6	O sistema oferece a possibilidade ao humano de vetar durante um determinado período de tempo e executa.
5	O sistema executa uma sugestão se o humano aprovar.
4	O sistema oferece uma alternativa.
3	O sistema descarta algumas da soluções obtidas.
2	O sistema oferece um conjunto de soluções ou acções alternativas.
1	O sistema não oferece assistência: o humano deve tomar todas as decisões e acções.

por dez níveis distintos. O nível mais elevado (dez), corresponde a um sistema absolutamente automatizado e que não possui qualquer tipo de interacção com o operador humano. O nível menos elevado (um), correspondem a sistemas que não oferecem qualquer tipo de assistência ao operador humano, sendo que é o operador humano o responsável por todas e quaisquer tomadas de decisão ou execução de acções.

Parasuraman et al [PW08] estudaram o impacto da utilização de vários níveis de automatização em diferentes sistemas e propuseram um modelo de tipos e níveis de automatização baseado na escala proposta por Sheridan & Verplank [SV78]. Nesse modelo identificam quatro níveis num sistema segundo tipos generalizados de funções: aquisição de informação, análise de informação, capacidade de decisão e execução de acção. Segundo os autores, estes níveis pode conter diferentes graus de automatização segundo o tipo de interacção com o humano. Ainda nesse estudo, os autores defendem que o desempenho de um qualquer sistema automatizado tende a aumentar quando engloba algum tipo de interacção humana. Uma das principais razões apontadas para este facto consiste na capacidade que o sistema adquire de se adaptar melhor face a situações imprevistas através da informação que lhe é fornecida pelo operador humano. Os autores concluem afirmando que, embora o modelo proposto ofereça uma boa orientação no projecto e implementação de automatização e interacção com o operador humano, não oferece nem pode oferecer regras e princípios exaustivos para a modelação desta interacção.

Agogino & Tumer [AT09] estudaram o impacto da utilização de um sistema multi-agente com ou sem a interacção de operadores humanos, no contexto de gestão de tráfego aéreo. Os agentes desse sistema são responsáveis pela apresentação de propostas que melhorem o tráfego aéreo, reduzindo o congestionamento. Os autores referem que a maior parte das soluções automatizadas para gestão de tráfego aéreo não consideram a interacção do operador humano com o sistema, e são da opinião que essa característica pode levar a soluções potencialmente perigosas de um ponto de vista técnico e politicamente difíceis de aceitar. Para colmatar esta falha, os autores propõem que sejam os operadores humanos a validar as propostas dos agentes. Não aceitando a solução

proposta pelo agente, o operador humano deve comunicar ao sistema uma solução para o problema calculada por si. Contudo, é assumido pelos autores que o objectivo do operador humano é reduzir o congestionamento do tráfego aéreo e que cada operador recebe um incentivo por seguir as propostas dos agentes. Do lado dos agentes, a interacção com o operador humano permite a cada agente perceber como fazer melhores propostas. Os agentes recebem uma recompensa de acordo com a validação ou rejeição feita pelo operador humano e tentam perceber, quando lhes é rejeitada a solução, qual o desfasamento que existe entre a solução proposta pelo sistema e a solução indicada pelo operador humano. Os autores concluem que o sistema multi-agente que interage com o operador humano permite aumentar a eficiência face ao desempenho dos operadores humanos. Outra importante conclusão é que os agentes do sistema aprendem melhor caso o operador humano não valide sempre as propostas que o sistema apresenta. Contudo, se o operador humano ignorar todas as propostas do sistema, o desempenho deste acaba por sofrer uma decréscimo acentuado.

Baxter & Horn [BH05] descrevem um sistema multi-agente que permite controlar uma equipa de UAV [4]. A função do sistema é identificar um tipo específico de corpos e disparar sobre eles, eliminando-os. Um dos principais objectivos do estudo era tentar reduzir a carga de trabalho ao operador humano. Contudo, o sistema deve continuar a fornecer informação suficiente ao operador humano de forma a que este consiga, em cada momento, perceber o que os agentes pensam sobre a situação actual. O inverso deve também ser verdade, ou seja, os agentes devem ser informados do ponto de vista do operador sobre a situação. Os autores descrevem que os resultados das experiências realizadas são favoráveis e que existiu uma boa sinergia entre a actividade do operador humano e a actividade dos agentes.

Dos exemplos apresentados se conclui a importância do operador humano num sistema automatizado, característica que será incluída no sistema MASDIMA, neste trabalho.

3.4 Resumo

Neste capítulo foi dada uma visão geral sobre o trabalho desenvolvido na área de sistemas multi-agente aplicados aos mais variados contextos. Foi visto que, dos sistemas automatizados aplicados à gestão de rupturas revistos, são poucos aqueles que consideram mais do que uma das perspectivas do problema. Nenhum dos sistemas existentes utiliza um paradigma multi-agente nem uma abordagem integrada como o MASDIMA. Assim, o estado da arte foi alargado a sistemas multi-agente que abordam o problema de implementação, quer de aprendizagem multi-agente, quer de interacção de um operador humano com um sistema automatizado. Verificou-se que nenhum dos exemplos descritos possui exactamente as mesmas características e dinâmica do MASDIMA, sobretudo quando comparada com a área de gestão de rupturas num centro de controlo operacional aéreo.

Dada a revisão do estado da arte que foi efectuada, conclui-se que o trabalho que se pretende executar pode demonstrar resultados interessantes, quer no campo de aprendizagem multi-agente,

[4] Veículos aéreos inabitados, do inglês *Uninhabited Air Vehicles*.

onde existem vários agentes a aprender e a interagir num ambiente simultaneamente competitivo e cooperativo, quer no campo da interacção de um operador humano com um sistema multi-agente.

A literatura revista forneceu um bom ponto de partida para a implementação de ambas as funcionalidades, pois permitiu identificar problemas, boas práticas e uma análise prévia aos impactos que podem advir da sua implementação.

Capítulo 4

Aprendizagem na Negociação Automática

O presente capítulo tem por objectivo apresentar a solução de aprendizagem desenvolvida, nesta dissertação, para a negociação automática entre agentes, do ponto de vista dos agentes *Manager* no sistema MASDIMA[1]. Ao longo do capítulo são apresentadas as dificuldades que surgiram durante o desenvolvimento da solução, as decisões tomadas e o racional por trás destas.

4.1 Introdução

A negociação automática utilizada no MASDIMA, como foi descrita no capítulo 2, é parte integrante do protocolo de interacção GQN[2] e decorre entre o agente *Supervisor* e os agentes *Manager*, que são responsáveis por calcular e enviar propostas de soluções integradas ao agente *Supervisor*. Este, por seu lado, deve avaliar todas as propostas recebidas segundo a sua função de utilidade U, representada na equação 2.2, e escolher a melhor desse ponto de vista.

O processo de aprendizagem na negociação automática foi implementado do lado dos agentes *Manager*. O objectivo do processo de aprendizagem é permitir aos agentes *Manager* efectuar propostas capazes de vencer a negociação por melhor se adequarem às preferências do agente *Supervisor*. No entanto os agentes *Manager* não possuem informação quanto às preferências do agente *Supervisor*, que são privadas. A única informação disponível é a avaliação atribuída pelo agente *Supervisor* aos atributos das propostas efectuadas em rondas anteriores e o facto de terem sido ou não os vencedores da ronda ou negociação. É através dessa informação que os agentes *Manager* se podem orientar.

[1] Do inglês *Multi-Agent System for Disruption Management*.
[2] Do inglês *Generic Q-Negotiation*.

Durante o desenvolvimento da solução foram implementadas e testadas versões distintas com diferenças substanciais. Depois de realizadas algumas experiências preliminares com a versão inicialmente projectada, observou-se que os resultados, embora positivos, não eram suficientemente satisfatórios. Procurou-se alterar a solução de forma a melhorar os resultados obtidos. A melhoria constatada levou ao abandono da versão inicial em detrimento da versão com melhores resultados. A versão final foi um produto de sucessivas alterações à versão inicial. Para entender as razões que levaram à implementação do mecanismo final é importante conhecer a primeira versão. Proceder-se-á à documentação de toda a evolução do desenvolvimento, procurando justificar devidamente as decisões que foram tomadas. A comparação de métricas entre as versões poderão ser encontradas no capítulo 6.

É no sub-capítulo 4.2 que se descrevem as duas versões implementadas. Na secção 4.2.1 descreve-se a versão inicial do mecanismo, enquanto que na secção 4.2.2 está descrito o mecanismo na sua versão final. No sub-capítulo 4.3 apresentam-se as configurações utilizadas para os algoritmos *Q-Learning* e *Sarsa*. No sub-capítulo 4.4 são resumidas algumas ideias fundamentais apresentadas neste capítulo.

4.2 Mecanismo de Aprendizagem

Mecanismo de aprendizagem será o termo utilizado para definir o processo que permite aos diferentes agentes *Manager* aprender a melhorar a avaliação atribuída pelo agente *Supervisor* às suas propostas.

Durante a revisão do estado da arte, seleccionou-se a aprendizagem por reforço como sendo o tipo de aprendizagem mais adequado ao mecanismo de aprendizagem na negociação automática, por comparação às aprendizagens supervisionada e não supervisionada. Os algoritmos de aprendizagem seleccionados foram o *Q-Learning* e *Sarsa*, cuja configuração pode ser encontrada no sub-capítulo 4.3. No capítulo 6 são apresentados alguns dados que permitem a comparação entre o desempenho de ambos os algoritmos.

Foram ainda identificadas, a partir da revisão do estado da arte, duas possíveis arquitecturas para o mecanismo de aprendizagem: aprendizagem por equipa ou aprendizagem concorrente. Na aprendizagem por equipa, existe um mecanismo de aprendizagem que é partilhado pelos vários agentes que constituem essa equipa. Já na aprendizagem concorrente cada agente possui o seu próprio mecanismo de aprendizagem. Um dos pilares do MASDIMA é a descentralização da complexidade do problema e, consequentemente, da informação. Nenhum dos agentes possui conhecimento completo sobre um problema. A aprendizagem por equipa exige uma certa centralização da informação, pelo que não parece ser a melhor alternativa ao cenário existente. Para além do mais, os agentes *Manager* estão a competir entre si. Se partilhassem o mesmo processo de aprendizagem, teriam a possibilidade de aceder a uma mesma estratégia, o que iria colidir contra os princípios da competição. Assim, a aprendizagem concorrente parece ser a melhor abordagem no contexto do problema que aqui se apresenta. Além de resolver os problemas anteriormente

referidos, proporciona ainda a vantagem de reduzir o espaço de estados do algoritmo, algo que é muito importante para que exista uma rápida convergência.

Na solução proposta cada agente *Manager* possui o seu próprio processo de aprendizagem por reforço que é baseado na avaliação qualitativa dada pelo agente *Supervisor* aos atributos que compõem a sua proposta de solução e no facto de o agente *Manager* ter vencido ou não a ronda com essa proposta.

Como já foi referido, o mecanismo final é uma evolução da primeira conceptualização, já depois de executadas alterações a diferentes níveis, como sejam a própria arquitectura do mecanismo ou as representações do estado e acção. Contudo, optou-se por dividir, de forma mais demarcada, dois mecanismos, *A* e *B*, segundo as modificações que o mecanismo sofreu a nível de arquitectura. Cada um destes mecanismos sofreu também alterações a nível interno, como será devidamente apresentado nos sub-capítulos que se seguem.

4.2.1 Mecanismo *A*

O mecanismo *A* consiste na solução inicialmente projectada e a sua arquitectura é apresentada na figura 4.1. O primeiro factor considerado na sua construção foi a existência de um ambiente simultaneamente cooperativo e competitivo. Os agentes *Manager* são obrigados a cooperar, embora compitam entre si para vencer a negociação, já que cada solução tem de conter um plano de domínio para cada perspectivas.

A avaliação dada pelo agente *Supervisor* a uma proposta integrada abrange todas as perspectivas, pelo que a solução nunca é avaliada apenas pela qualidade da solução parcial da perspectiva do agente *Manager* que a propôs. Contudo, quando um dos agentes, seja o agente *Aircraft Manager*, pede aos dois outros agentes *Manager* uma solução parcial que complete a sua, segundo o funcionamento do protocolo GQN, impõe restrições à partida que os agentes *Crew Member Manager* e *Passenger Manager* não podem desrespeitar, de forma a que as dependências entre as perspectivas sejam garantidas. Os agentes a quem foi pedido que completassem a solução têm de o fazer de acordo com a solução parcial já definida pelo agente que pediu essa colaboração. Deste ponto de vista, os agentes *Crew Member Manager* e *Passenger Manager* não têm tanta responsabilidade sobre a proposta quanto o agente *Aircraft Manager*, pelo que não seria justo atribuir a mesma recompensa aos três agentes, seja ela boa ou má. Qualquer que seja a recompensa, esta deve ser atribuída de acordo com o peso de responsabilidade que o agente teve sobre a proposta. É também injusto considerar com o mesmo peso a avaliação dada a propostas onde o agente colaborou sob restrições e propostas que apresentou ao *Supervisor* e influenciar todo o processo de aprendizagem sem fazer qualquer distinção entre estas.

O agente *Manager* não está na mesma situação quando está a lidar directamente com o agente *Supervisor* ou quando está a responder a um outro agente *Manager*. Pelas razões supracitadas, e de forma a não misturar os diferentes contextos em que o agente se encontra, implementaram-se três motores[3] de aprendizagem distintos e concorrentes por cada agente *Manager*.

[3]Um motor representa a execução de uma instância de um algoritmo. Cada motor mantém o seu próprio espaço de estados e acções possíveis que é actualizado segundo diferentes partes da avaliação atribuída e representada em 2.3.

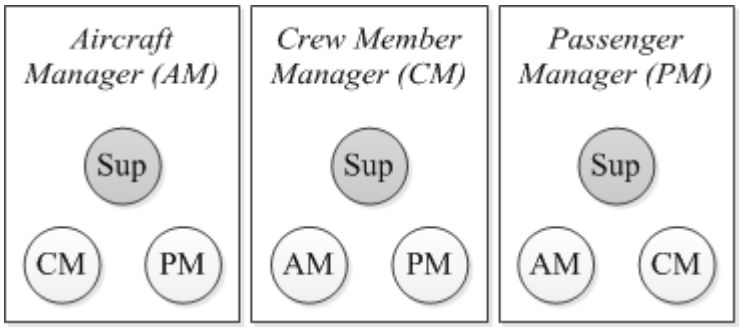

Figura 4.1: Arquitectura do Mecanismo de Aprendizagem *A* - Cada círculo representa um motor distinto e concorrente. Os motores *Sup* são destinados à negociação directa com o agente *Supervisor*, enquanto que os motores *AM*, *CM* e *PM* são destinados à negociação entre agentes *Manager*.

Um dos motores diz respeito à negociação que este tipo de agente tem directamente com o agente *Supervisor*. Numa ronda, todos os agentes *Manager* devem apresentar uma proposta integrada ao agente *Supervisor*. Quando recebem a avaliação da proposta que apresentaram, cada agente *Manager* selecciona a avaliação dos atributos da sua própria perspectiva e é com esta avaliação que actualiza o motor referido. A avaliação que diz respeito a atributos de outras perspectivas é reencaminhada para os agentes responsáveis, que deverão actualizar o motor de aprendizagem destinado à negociação com o agente que lhes pediu que completassem a sua solução e que agora reencaminha a avaliação atribuída à parte que lhes diz respeito.

Como já foi referido no capítulo 3, a aprendizagem por reforço utiliza os conceitos de estado e acção. Torna-se assim necessário definir estes conceitos no contexto específico do problema, para que seja possível compreender o fluxo de todo o processo de aprendizagem.

4.2.1.1 Definição de estado, acção e recompensa

O objectivo do mecanismo de aprendizagem é ajudar o agente que o implementa a adaptar-se às preferências do agente *Supervisor*, que lhe são desconhecidas. Contudo, as preferências do agente *Supervisor* dependem do problema em negociação.

Para que o agente *Manager* possa adaptar as suas propostas em função do problema em negociação é necessário que a representação do estado contenha atributos que permitam caracterizar o problema. Um problema pode ser caracterizado por inúmeros aspectos, de entre os quais se destacam a causa que originou o problema, os recursos que foram afectados por esse mesmo problema incluindo o número de passageiros e os seus destinos, a origem e destino dos voos, a hora de chegada e partida, entre outros.

Como referido no capítulo 3, os algoritmos *Q-Learning* e *Sarsa* efectuam uma pesquisa sobre o espaço de estados em busca do par estado-acção que poderá maximizar a utilidade do agente. Assim, o espaço de estados não deverá ser demasiado grande, pois tal característica terá consequências negativas a nível do desempenho do algoritmo. Por essa razão, dos aspectos que podem caracterizar o problema seleccionaram-se apenas dois: a causa do problema original e o recurso

afectado. As treze principais causas consideradas, que podem ser consultadas na tabela A.1 do anexo A, foram identificadas a partir de entrevistas efectuadas aos membros responsáveis pela resolução destes problemas no centro de controlo operacional da TAP Portugal. Quando é originado um problema, a causa comunicada aos diferentes agentes *Manager* é a mesma, pelo que todos partilham este conhecimento. O mesmo não acontece com o recurso afectado, que varia consoante a perspectiva do problema. Uma vez que um agente *Manager* possui apenas conhecimento suficiente para resolver problema relacionados com a sua perspectiva, não poderá fornecer uma solução cujo recurso afectado ultrapasse as fronteiras do seu domínio. No caso da perspectiva *Aircraft*, esse atributo será preenchido com o modelo do avião afectado como, por exemplo, A330 (Airbus 330), sendo que existem cinco modelos distintos. Na perspectiva *Crew Member* o atributo terá o valor do tipo do corpo do avião afectado que poderá ser WB (*Wide Body*) ou NB (*Narrow Body*). Na perspectiva *Passenger* o atributo poder ter o valor BC (*Business Class*) ou YC(*Economic Class*).

Contudo, a caracterização do problema não é suficiente para que o agente consiga perceber se esse estado espelha a situação em que se encontra. Uma vez que se pretende que o agente *Manager* adapte a sua proposta às preferências do agente *Supervisor*, o estado deve espelhar ainda a avaliação atribuída pelo agente *Supervisor* à sua última proposta.

Com base no que foi anteriormente referido, a representação de um estado para cada agente *Manager* segue o n-tuplo da expressão 4.1.

$$e =< cause, resource, class_{Cost}, class_{Delay} > \tag{4.1}$$

O elemento *cause* representa a causa que originou o problema e o elemento *resource* o recurso afectado. Já os elementos $class_{Cost}$ e $class_{Delay}$ correspondem à classificação qualitativa dada pelo agente Supervisor aos atributos *Cost* e *Delay*[4] pertencentes à perspectiva do agente. Os valores destes elementos poderão ser *Low*, *Ok*, *High* ou *VeryHigh*, como descrito no capítulo 2. O espaço de estados tem diferentes dimensões de acordo com a perspectiva em questão. Na perspectiva *Aircraft*, o espaço de estados poderá ser constituído, no máximo, por 832[5] estados. Ambas as perspectivas *Crew Member* e *Passenger* poderão ter, no máximo, um espaço de estados com 416[6] estados.

Assim como o estado, a acção possui elementos respeitantes à avaliação atribuída e ao domínio do problema e está representada no n-tuplo da expressão 4.2.

$$a =< action_{Domain}, action_{Cost}, action_{Delay} > \tag{4.2}$$

Os elementos $action_{Cost}$ e $action_{Delay}$ representam as acções a tomar em relação à classificação dada aos atributos *Cost* e *Delay*, respectivamente. Os valores possíveis, bem como o seu significado, estão expressos na tabela A.5 do anexo A.

[4] *TripTime* no caso da perspectiva *Passenger*.

[5] Valor obtido a partir da multiplicação da quantidade de valores possíveis para cada elemento que constitui um estado. O elemento *cause* pode ter 13 valores, *resource* pode ter 4 valores, $class_{Cost}$ 4 valores e $class_{Delay}$ também 4 valores, o que perfaz $13 \times 4 \times 4 \times 4 = 832$ estados.

[6] $13 \times 2 \times 4 \times 4 = 416$ estados

O elemento $action_{Domain}$ representa o plano de domínio do problema e varia consoante a perspectiva. As tabelas A.3, A.4 e A.2 do anexo A ilustram os valores possíveis do elemento para cada perspectiva, bem como a descrição do significado de cada plano.

Considerando o espaço de estados, a quantidade de acções existentes a partir de cada estado varia também consoante a perspectiva considerada. Na perspectiva *Aircraft* cada estado poderá ter, no máximo, 36[7] acções. Na perspectiva *Crew Member*, a quantidade de acções disponíveis a partir de um estado poderá atingir, no máximo, 72[8] acções. Na perspectiva *Passenger*, por estado, poderão existir 27[9] acções diferentes.

Cada solução devolvida pelo agente *Specialist* ao agente *Manager* contém um atributo que corresponde ao plano de domínio, pelo que o agente *Manager* deve filtrar todas as soluções parciais segundo o plano de domínio devolvido pelo processo de aprendizagem. Todas as soluções que não respeitem esse plano de domínio são automaticamente rejeitadas.

Na primeira execução do algoritmo, as tabelas dos valores Q de cada par estado-acção são iniciadas com o valor 0.0[10]. A actualização do valor Q de cada par estado-ação depende do algoritmo em utilização ($Q-Learning$ ou *Sarsa*) e é feita segundo as equações 3.1 ou 3.2, onde R é a função de recompensa definida em 4.3 e pen_i, definida em 4.4, é a penalização associada à classificação do atributo i. O valor da penalização a atribuir, em 4.4, foi determinado de forma empírica, sendo que se tentou associar valores mais altos de penalização a piores classificações.

No cálculo da recompensa do mecanismo A são apenas considerados os atributos da perspectiva do agente ao qual pertence o mecanismo de aprendizagem. Como cada agente *Manager* é responsável por dois atributos da proposta, o valor da constante m da equação 4.3 é 2 em qualquer dos mecanismos.

$$R = \begin{cases} m & \text{se vencedor,} \\ \frac{m}{2} - \sum_{i=1}^{m} pen_i & \text{se perdedor.} \end{cases} \qquad m \text{ - número de atributos utilizados} \tag{4.3}$$

$$pen_i = \begin{cases} 2.0 & \text{se Very High,} \\ 1.5 & \text{se High,} \\ 0.5 & \text{se Low,} \\ 0.0 & \text{se Ok.} \end{cases} \tag{4.4}$$

4.2.1.2 Processo de Aprendizagem

Depois de definido o conceito de estado, acção e recompensa, interessa agora descrever o ciclo do processo de aprendizagem. A descrição será feita, a título de exemplo, apenas para o agente

[7] $4 \times 3 \times 3 = 12$ acções

[8] $8 \times 3 \times 3 = 72$ acções

[9] $3 \times 3 \times 3 = 27$ acções

[10] Inicialmente considerou-se a hipótese de iniciar os valores Q de cada par estado-acção tendo em conta uma tabela de probabilidades extraída de entrevistas aos membros do controlo operacional aéreo da TAP Portugal. Essas probabilidades traduziam uma relação entre a causa que deu origem ao problema (representada no elemento *cause* de um estado) e o plano de domínio que mais frequentemente é aplicado nessa situação (representada pelo elemento $action_{Domain}$ de uma acção). Contudo, este procedimento foi abandonado visto que essas probabilidades eram calculadas tendo em conta o comportamento do operador humano, e poderiam viciar todo o mecanismo de aprendizagem.

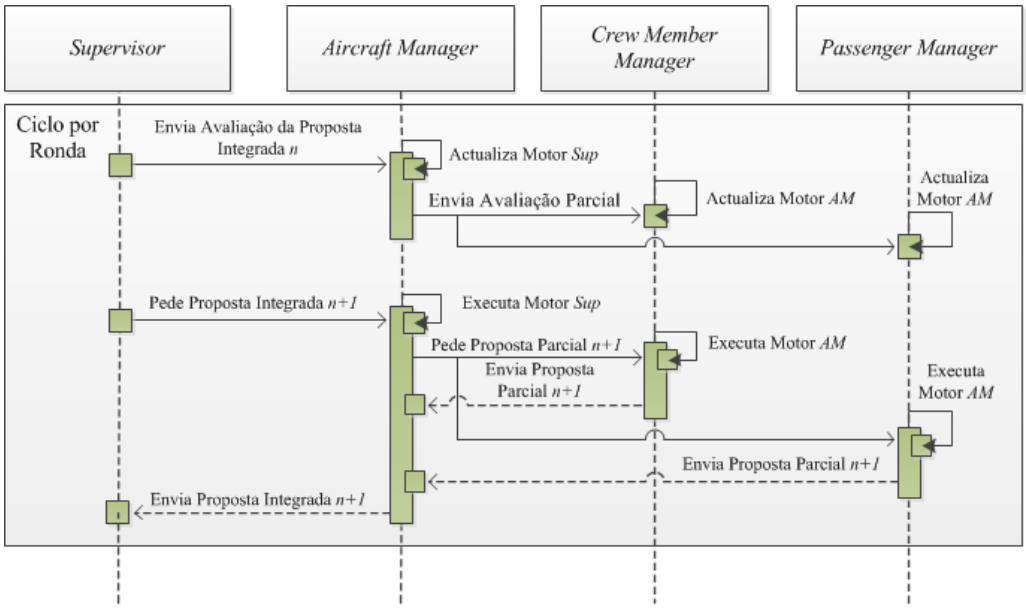

Figura 4.2: Protocolo GQN com Mecanismo de Aprendizagem *A* - Diagrama de sequência que representa o processo de aprendizagem do mecanismo *A* embebido no protocolo GQN, apenas para o agente *Aircraft Manager*.

Aircraft Manager. No entanto, o processo para os outros agentes *Manager* é idêntico e processa-se em paralelo. Note-se que, na primeira ronda, nenhum agente *Manager* possui qualquer tipo de avaliação, pelo que a primeira proposta é feita sem recurso ao algoritmo de aprendizagem. O processo de aprendizagem começa no final de cada ronda, quando o agente *Supervisor* comunica a avaliação a todos os agentes *Manager* que propuseram uma solução integrada, como representado graficamente no diagrama da figura 4.2.

A descrição do processo será divida em duas partes, de forma a facilitar a sua compreensão. Na primeira é descrito o ciclo de aprendizagem executado quando o agente *Manager* interage directamente com o agente *Supervisor*, pelo que não é mencionada a actividade dos restantes agentes *Manager*. A descrição do comportamento de um agente *Manager* quando este colabora na construção de uma solução parcial de outro agente *Manager* constitui a segunda parte.

Ciclo de aprendizagem quando é o agente *Aircraft Manager* a negociar directamente com o agente *Supervisor*:

1. O agente *Supervisor* comunica a avaliação dada à última proposta do *Aircraft Manager*. O agente *Aircraft Manager* comunica a avaliação parcial aos restantes agentes *Manager* e actualiza o espaço de estados do motor *Sup* (figura 4.2).

2. O agente *Supervisor* convoca o *Aircraft Manager* para a próxima ronda. Este último procederá da seguinte forma:

- Se tiver sido o vencedor na última ronda, selecciona a acção com o mesmo plano de domínio da acção anterior, mas com o valor *Keep* para os restantes elementos da acção, já que na próxima ronda enviará a mesma proposta. Segue para o passo 7.
- Se tiver perdido a ronda, seleccionará uma nova acção de acordo com a política definida pelo algoritmo.

3. O agente *Aircraft Manager* convoca o agente *Specialist* para que este lhe forneça uma lista de soluções parciais para a sua perspectiva.

4. Ao receber essa lista filtra-a de acordo acção seleccionada.

   - Se existirem soluções que respeitam a acção, passa para o passo seguinte.
   - Se nenhuma solução satisfizer a acção escolhida, selecciona uma nova acção e repete este passo. Existe um limite quanto ao número de vezes que esta situação pode acontecer. Se esse limite for ultrapassado, o agente *Aircraft Manager* não selecciona qualquer ação e comunica ao agente *Supervisor* que falhou. Neste caso, a próxima ronda funciona como se fosse a primeira.

5. O agente *Aircraft Manager* ordena pelo critério de maior utilidade a lista resultante do passo anterior e escolhe a primeira da lista.

6. O agente *Aircraft Manager* envia a solução escolhida, e as restrições inerentes para que os outros agentes *Manager* a completem. Podem acontecer as seguintes situações:

   - Os agentes *Manager* conseguem completar a solução, pelo que o agente *Aircraft Manager* passa para o próximo passo.
   - Um ou ambos os agentes *Manager* não conseguem completar a solução, pelo que o agente *Aircraft Manager* deve seleccionar a próxima solução da lista ordenada e voltar ao início deste passo. Caso já não existam mais soluções disponíveis, o agente *Aircraft Manager* comunica ao agente *Supervisor* que falhou, terminando o ciclo. Neste caso, a próxima ronda funciona como se fosse a primeira.

7. O agente *Aircraft Manager* reúne as soluções parciais numa solução integrada e envia-a ao agente *Supervisor* para avaliação. Retorna ao passo 1.

Ciclo de aprendizagem quando é o agente *Aircraft Manager* a receber uma convocatória por parte do agente *Passenger Manager*:

1. O agente *Passenger Manager* comunica ao *Aircraft Manager* a avaliação dada aos atributos deste na proposta da última ronda. O agente *Aircraft Manager* actualiza o espaço de estados do motor *PM* (figura 4.2).

   - Se ganhar a última ronda, selecciona a acção com o mesmo plano de domínio da anterior mas com o valor *Keep* para os restantes atributos.

- Se perder a ronda, o agente fica à espera de uma nova convocatória, no passo 2.

2. O agente *Aircraft Manager* recebe uma convocatória do agente *Passenger Manager* que lhe envia a sua solução parcial e as restrições associadas. O agente *Aircraft Manager* selecciona uma nova acção de acordo com a política definida pelo algoritmo.

3. O agente O agente *Aircraft Manager* convoca os seus agentes *Specialist* para que estes lhe forneçam uma lista de soluções parciais para a sua perspectiva, de acordo com a solução parcial recebida.

4. Ao receber essa lista filtra-a de acordo acção seleccionada.

   - Se existirem soluções que respeitam a acção, segue para o passo 5.
   - Se nenhuma solução satisfizer a acção escolhida, selecciona uma nova acção e repete este passo. Existe um limite quanto ao número de vezes que esta situação pode acontecer. Se esse limite for ultrapassado, o agente não selecciona qualquer ação e comunica ao agente *Passenger Manager* que falhou, terminado o ciclo. Neste caso, a próxima ronda funciona como se fosse a primeira.

5. O agente *Aircraft Manager* ordena pelo critério de maior utilidade a lista resultante do passo anterior e escolhe a primeira dessa lista.

6. O agente *Aircraft Manager* envia a solução escolhida para o agente *Passenger Manager*.

A partir de resultados de experiências preliminares, foi possível detectar que, em grande número de rondas, os agentes *Manager* não eram capazes de apresentar nenhuma proposta integrada ao agente *Supervisor*. O agente *Manager* que ganhava a negociação na primeira ronda tendia a ser o vencedor de todas as rondas e, consequentemente, de toda a negociação.

Numa primeira análise aos resultados obtidos identificou-se uma possível causa: a existência do elemento $action_{Domain}$ que compõe uma acção do algoritmo de aprendizagem. Os agentes *Manager* filtram, a partir de um conjunto de soluções possíveis, aquelas que estão de acordo com a acção seleccionada pelo algoritmo. Era frequente acontecer que, do conjunto de soluções possíveis, nenhuma respeitava o elemento $action_{Domain}$ da acção seleccionada. O agente ficava sem hipóteses de apresentar uma solução nessa ronda. O agente vê-se impossibilitado de utilizar o algoritmo de aprendizagem para propor uma solução ao *Supervisor* nessa ronda, pois não recebeu qualquer tipo de avaliação, e utiliza apenas o critério da maior utilidade para escolher entre as soluções do conjunto. Só depois de decorrida esta nova ronda pode voltar a utilizar o algoritmo. Todavia, o resultado tendia a ser o mesmo e nenhuma solução do conjunto obtido respeitava a nova acção seleccionada. Com este comportamento, os agentes raramente conseguiam utilizar o algoritmo de aprendizagem. Mesmo o agente que ganhava a primeira ronda nunca chegava a utilizar o mecanismo de aprendizagem para devolver uma solução. A primeira solução apresentada tinha sido construída sem recurso à aprendizagem. Como raramente se conseguia destituir esta solução, o agente apresentava sempre a mesma solução.

A partir desta hipótese alterou-se a representação da acção, que anteriormente seguia o n-tuplo da expressão 4.2. A acção do mecanismo *A* passou a seguir o n-tuplo da expressão 4.5, sendo que o elemento $action_{Domain}$ foi removido.

$$a =< action_{Cost}, action_{Delay} > \tag{4.5}$$

Após esta alteração, verificou-se uma diminuição significativa no número de rondas em que os agentes *Manager* não conseguiam apresentar uma proposta. Embora a alteração não permitisse anular completamente esse número, a melhoria verificada levou à permanência dessa representação de uma acção para todos os motores do mecanismo *A*.

A remoção do parâmetro $action_{Domain}$ teve ainda outro impacto a nível da dimensão do espaço de estados. Em todas as perspectivas, a dimensão do conjunto de acções disponíveis a partir de um estado diminui para 9[11]. Tal facto permitiu acelerar a convergência do algoritmo.

Uma segunda análise permitiu formular uma outra hipótese que podia explicar a dificuldade dos agentes em apresentar uma proposta. Tome-se, a título de exemplo, o agente *Aircraft Manager*. Quando este agente inicia a negociação com os outros agentes *Manager*, embora o agente *Aircraft Manager* consiga escolher uma solução parcial, muitas vezes são os seu colaboradores que não lhe conseguem responder com uma solução parcial, inibindo o agente *Aircraft Manager* de completar uma solução que possa apresentar ao agente *Supervisor*. Este comportamento poderá estar associado os motores destinados à negociação entre os agentes *Manager*. O objectivo destes motores (*AM*, *CM* e *PM*) passava por não misturar os diferentes contextos de negociação onde os agentes se viam envolvidos. Contudo, em conjunto com as restrições que já eram impostas aos agentes *Crew Member Manager* e *Passenger Manager*, acabavam por ter um efeito muito restritivo na escolha de uma solução. Muitas vezes tornavam os agentes incapazes de devolver uma solução parcial ao agente *Aircraft Manager* que respeitasse as restrições e acção seleccionada em simultâneo [TCRO13].

Foi a partir desta última hipótese que se evoluiu o mecanismo *A* para uma nova versão, com uma arquitectura diferente e com uma linha de pensamento distinta. O novo mecanismo, denominado Mecanismo *B*, é apresentado na próxima secção.

4.2.2 Mecanismo *B*

A origem do mecanismo *B* está ligada às experiências preliminares feitas com o mecanismo *A*, as quais permitiram detectar pontos negativos na utilização do processo de aprendizagem. Os motores do mecanismo *A* destinados à negociação entre agentes *Manager*, em conjunto com as restrições que já eram impostas pelos agentes nessa mesma negociação, tornava todo o processo demasiado restritivo, dificultando e, muitas vezes, impossibilitando a apresentação de propostas por parte dos agentes *Manager* ao agente *Supervisor*.

A característica que melhor distingue o mecanismo *B* do *A* é a inexistência dos motores destinados à negociação entre os agentes *Manager*. Como demonstrado na figura 4.3, cada agente

[11] $3 \times 3 = 9$ acções

Figura 4.3: Arquitectura do Mecanismo de Aprendizagem *B* - Cada círculo representa um motor distinto e concorrente. Todos os motores são destinados à negociação directa com o agente *Supervisor*, por oposição à arquitectura representada na figura 4.1.

Manager fica responsável por um único motor de aprendizagem que é actualizado segundo a avaliação directamente proveniente do agente *Supervisor* e que diz respeito apenas às propostas efectuadas pelo agente *Manager* em questão. Neste mecanismo, o objectivo é fazer com que o agente *Manager* que pede a colaboração dos outros *Manager*, impondo-lhes restrições, tenha mais consciência das capacidades dos seus colaboradores e que aja de acordo com estas [TCRO13].

A responsabilidade de completar a solução deixa de estar apenas sobre os agentes *Manager* que a completam. Acredita-se que este mecanismo é mais adequado a um ambiente simultaneamente competitivo e cooperativo, como é o dos agentes *Manager*. Se o agente quer ganhar a negociação e, ao mesmo tempo, necessita da cooperação dos agentes que consigo competem, então deve tornar-lhes a tarefa de o auxiliar mais simples.

4.2.2.1 Definição de estado, acção e recompensa

A representação da acção adoptada é a versão já evoluída da representação de uma acção no mecanismo *A*, expressa na fórmula 4.5. A representação do estado sofreu alterações de forma a coincidir com a filosofia deste novo mecanismo. No mecanismo *A*, atribuía-se a responsabilidade de melhorar a solução parcial aos agentes a quem essa solução era requisitada. O agente que requisitava a solução e que impunha restrições apenas se adaptava à classificação que dizia respeito aos atributos da sua perspectiva, não tendo nenhuma consideração pelas capacidades dos agentes a quem pedia colaboração. A própria representação do estado, expressa na fórmula 4.2, não lhe dava qualquer noção da avaliação atribuída aos atributos das perspectivas dos seus colaboradores.

No mecanismo *B*, a representação do estado passou a depender de mais um elemento, como é possível ver a partir do n-tuplo da expressão 4.6, denominado $global\_situation$.

$$e =< cause, resource, class_{Cost}, class_{Delay}, global\_situation > \tag{4.6}$$

O elemento $global\_situation$ sintetiza a classificação dada aos atributos das restantes perspectivas. Poder-se ia, desde logo, ter definido a representação de um estado com um elemento que correspondesse a cada atributo que compõe uma solução. Contudo, essa representação iria aumentar de forma muito elevada o espaço de estados de cada motor e atrasar a convergência do algoritmo, o que se pretende evitar.

O intuito do elemento *global_situation* é tornar o agente mais consciente da qualidade global da proposta e da situação dos agentes que com ele colaboram sem, no entanto, sobrecarregar os seus colaboradores [TCRO13]. Este elemento, fazendo parte do estado, é também calculado quando o agente *Manager* recebe a avaliação proveniente do agente *Supervisor*, mas através da avaliação qualitativa atribuída aos atributos das perspectivas sobre as quais não possui qualquer conhecimento (todas, excepto a sua própria). Para calcular o valor qualitativo do elemento *global_situation*, atribui-se uma pontuação quantitativa a cada valor de classificação qualitativa existente, de acordo com a tabela A.6 do anexo A.

Após somar os pontos associados a cada atributo, o valor de *global_situation* será calculado segundo a equação 4.7, podendo o elemento ter um de três valores qualitativos distintos: *VeryBad*, *Bad* e *Good*. O intervalo quantitativo entre cada um destes valores foi determinado de forma empírica.

$$global\_situation = \begin{cases} \text{Good} & \text{se } 0 \geq \text{pontuação} \leq 2, \\ \text{Bad} & \text{se } 2 > \text{pontuação} \leq 4, \\ \text{Very Bad} & \text{se pontuação} > 4. \end{cases} \tag{4.7}$$

Através dessa informação, pretende-se que o agente consiga não apenas perceber quais são as melhores propostas de um ponto de vista individual, mas também perceber quais são as suas próprias propostas de solução parcial, e respectivas restrições, que podem contribuir para melhorar as soluções parciais dos seus colaboradores.

Ao ter resumido num elemento informação referente a quatro atributos distintos, o espaço de estados do mecanismo *B* passou a ter 2496[12] estados. Se se tivesse incluído os quatros elementos separadamente o espaço de estados teria 212992[13] estados, o que tornaria mais difícil uma rápida convergência do algoritmo.

Existe ainda outra diferença entre os mecanismo *A* e *B*, referente à recompensa atribuída ao agente, que no mecanismo *A* era calculada segundo a equação 4.3. Uma vez que o estado do agente *Manager*, no mecanismo *B*, depende da avaliação atribuída aos atributos das outras perspectivas, a recompensa que o agente recebe passa a ser calculada tendo em conta a classificação de todos os atributos de todas as perspectivas. Contudo, os agentes não possuem conhecimento sobre todas as perspectivas e não são, por isso, os únicos responsáveis pela classificação obtida. De forma a não responsabilizar cada agente em demasia, associou-se um factor de desconto aos atributos que não pertencem à sua perspectiva e que tentam ter em consideração o grau de responsabilidade do agente. A nova equação que permite calcular a recompensa está representada em 4.8, onde $discount_i$, na equação 4.9, representa o factor de desconto associado aos atributos que não pertencem à perspectiva do agente *Manager*. O valor deste factor, no caso em que o atributo não pertence à perspectiva, foi determinado de forma empírica: começou por ser 1.0, mas o seu valor foi sendo diminuído após algumas experiências preliminares.

[12] Valor obtido a partir da multiplicação da quantidade de valores possíveis para cada elemento que constitui um estado. O elemento *cause* pode ter 13 valores, *resource* pode ter 4 valores, $class_{Cost}$ 4 valores, $class_{Delay}$ também 4 valores e *global_situation* 3 valores, o que perfaz $13 \times 4 \times 4 \times 4 \times 3 = 2496$ estados.

[13] $13 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 = 212992$ estados.

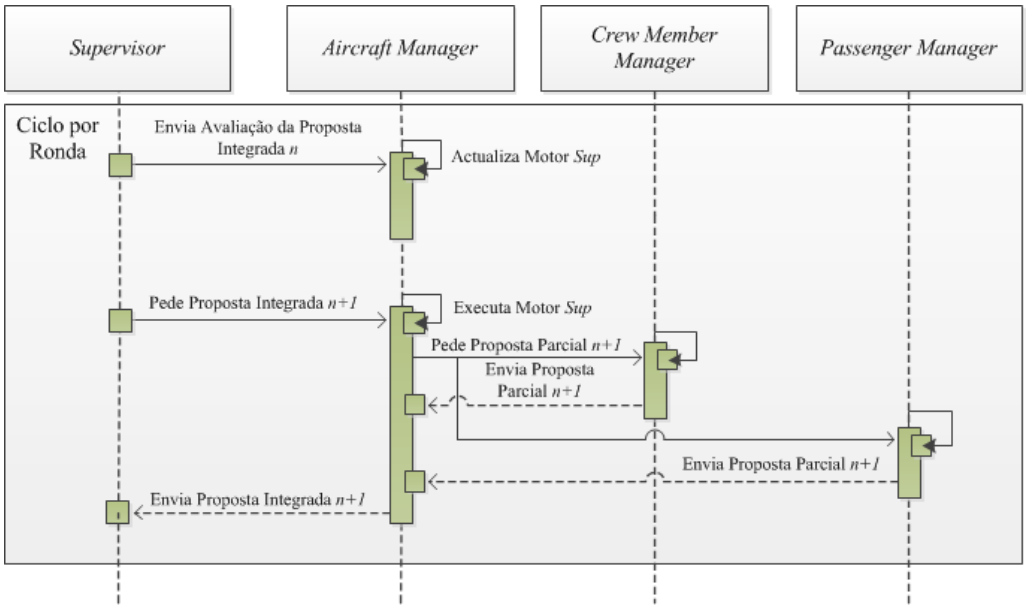

Figura 4.4: Protocolo GQN com Mecanismo de Aprendizagem *A* - Diagrama de sequência que representa o processo de aprendizagem do mecanismo *B*, para o agente *Aircraft Manager*, embebido no protocolo GQN.

Uma vez que são considerados os atributos de todas as perspectivas no cálculo da recompensa utilizado pelo mecanismo *B*, o valor da constante *m* da equação 4.8 é 6.

$$R = \begin{cases} m & \text{se vencedor,} \\ \frac{m}{2} - (\sum_{i=1}^{m} pen_i \times discount_i) & \text{se perdedor} \end{cases} \qquad m \text{ - número de atributos utilizados} \tag{4.8}$$

$$discount_i = \begin{cases} 1.0 & \text{se pertence à perspectiva,} \\ 0.4 & \text{se não pertence à perspectiva.} \end{cases} \tag{4.9}$$

4.2.2.2 Processo de Aprendizagem

O diagrama da figura 4.4 ilustra o processo de aprendizagem com o mecanismo *B* em conjunto com o protocolo GQN. Dos ciclos de aprendizagem descritos na secção 4.2.1 apenas se mantém aquele que diz respeito à negociação directa do agente *Manager* com o agente *Supervisor* embora ligeiramente alterado no ponto 1. O agente *Manager* que recebe a avaliação da solução integrada não a comunica aos agentes colaboradores. Apresenta-se, de seguida, o ponto 1 reformulado de acordo com o funcionamento do mecanismo *B*:

1. O agente *Supervisor* comunica a avaliação dada à última proposta do *Aircraft Manager*. O agente *Aircraft Manager* actualiza o espaço de estados do motor *Sup*.

O ciclo de aprendizagem entre os agentes *Manager* não se aplica, visto que o mecanismo *B* não possui motores de aprendizagem para a negociação entre os agentes *Manager*. Quando for

necessário interagir, na negociação entre agentes *Manager*, os agentes adoptam o protocolo GQN sem aprendizagem descrito no capítulo 2.

4.3 Algoritmos de Aprendizagem

Os algoritmos utilizados na implementação dos mecanismos já foram apresentados no capítulo 3: *Q-Learning* e *Sarsa*. Ambos os algoritmos utilizam uma estratégia de selecção de acções que não está definida, podendo variar de acordo com as preferências de quem implementa o algoritmo.

Das estratégias de selecção de uma acção apresentadas no capítulo 3, seleccionou-se a estratégia $Softmax$ em detrimento da estratégia aleatória $\varepsilon - greedy$. Após algumas experiências preliminares com ambos os mecanismos, verificou-se que uma distribuição mais uniforme das probabilidades de se seleccionar uma acção promovia mais visitas a estados com um valor Q mais baixo, tendo um certo impacto na convergência do algoritmo. Na estratégia aleatória $Softmax$ havia um desequilíbrio muito maior entre a probabilidade atribuída, que variava de acordo com o valor Q do par. Esse desequilíbrio tinha a sua origem na equação de actualização do valor do parâmetro t, representada na equação 4.10. O parâmetro t é utilizado pela estratégia *Softmax* na distribuição de *Boltzmann* (equação 3.3).

$$t = t - 0.9^n, \quad t_{inicial} = 12, \quad n\text{- número de iterações} \tag{4.10}$$

Ao decrescer o parâmetro t, as probabilidades de escolha de uma acção favoreciam os estados mais promissores. Tal facto resultava numa convergência mais rápida do espaço de estados. Embora não tivessem sido detectadas diferenças significativas entre as duas estratégias, em termos de tempo de convergência, optou-se pela utilização da estratégia $Softmax$ em ambos os algoritmos.

O valor inicial do parâmetro α das equações 3.1 e 3.2 corresponde a 0.9. Esse mesmo valor é actualizado segundo a equação 4.11.

$$\alpha = \frac{\alpha_{inicial}}{1 + \beta * n}, \quad n \text{ - número de iterações} \tag{4.11}$$

A constante β da equação 4.11 tem o valor de 0.05 e permite fazer decrescer mais lentamente, em cada iteração n, o valor de α. O seu valor foi determinado de forma empírica. O parâmetro γ das equações 3.1 e 3.2 tem o valor de 0.6.

Os valores utilizados no algoritmo têm por base o estudo de Rocha [Roc01], já citado no capítulo 3. Durante as experiências preliminares com ambos os mecanismos, estes valores revelaram-se adaptados ao contexto de aprendizagem na negociação automática entre agentes do sistema MASDIMA, pelo que foram preservados.

4.4 Resumo

Neste capítulo foi apresentada a solução desenvolvida para o problema identificado no capítulo 2.3. A solução final implementada não foi a inicialmente projectada. Factores como os resultados

de experiências preliminares e a própria maturação do assunto levaram ao abandono das versões iniciais da solução em detrimento de versões que auguravam melhores resultados.

Foram assim descritos dois mecanismos de aprendizagem distintos. O mecanismo *A*, descrito na secção 4.2.1, constitui a primeira solução desenvolvida. Com este mecanismo pretendia-se que o agente a quem era pedida a solução soubesse adaptar-se o melhor possível às exigências do contexto. Embora este mecanismo tenha sofrido algumas modificações a nível interno, foram as alterações à arquitectura do mecanismo *A*, e à filosofia inerente, que conduziram a um novo mecanismo *B*, descrito na secção 4.2.2. No mecanismo *B* são os próprios agentes que pedem a cooperação que devem ter mais consciência das capacidades dos seus colaboradores. As modificações efectuadas e a sua justificação acompanham toda a descrição de ambos os mecanismos.

Capítulo 5

Human-in-the-Loop

Neste capítulo é apresentada a funcionalidade desenvolvida, nesta dissertação, que permite a interacção do operador humano com o sistema MASDIMA[1]. Esta funcionalidade é denominada *Human-in-the-Loop*.

5.1 Introdução

No sub-capítulo 3.3 discutiu-se a viabilidade de permitir a um humano influenciar um sistema cujo objectivo é automatizar um dado processo. No contexto específico do sistema MASDIMA, a funcionalidade desenvolvida posiciona o sistema a meio da escala da tabela 3.1, o que irá permitir ao operador humano aprovar ou rejeitar a solução que o sistema devolve no fim de uma negociação automática. O operador humano não pode, no entanto, intervir durante o processo de negociação automática entre os agentes. Se o operador humano rejeitar a solução, o sistema deve iniciar uma nova negociação automática.

A renegociação do problema tem em vista a procura de uma solução que se adapte às exigências do contexto real, representadas pelas preferências do operador humano. O sistema só será capaz de ir de encontro às preferências do operador humano se lhe for fornecido algum tipo de informação em relação à solução encontrada. Assim, é necessário integrar no sistema uma interface que permita ao operador humano avaliar a solução devolvida pelo sistema. Uma das características essenciais a considerar quanto a este tópico é a necessidade de apresentar informação relevante ao operador humano que lhe permita ter uma boa percepção da situação actual, e assim, agir em conformidade com a situação.

O protótipo do sistema MASDIMA já possuía, numa versão anterior ao início do desenvolvimento desta dissertação, uma interface (representada na figura 2.3) que permitia ao operador humano monitorizar quer o plano operacional, quer quaisquer rupturas e problemas que possam

[1] Do inglês *Multi-Agent System for Disruption Management*.

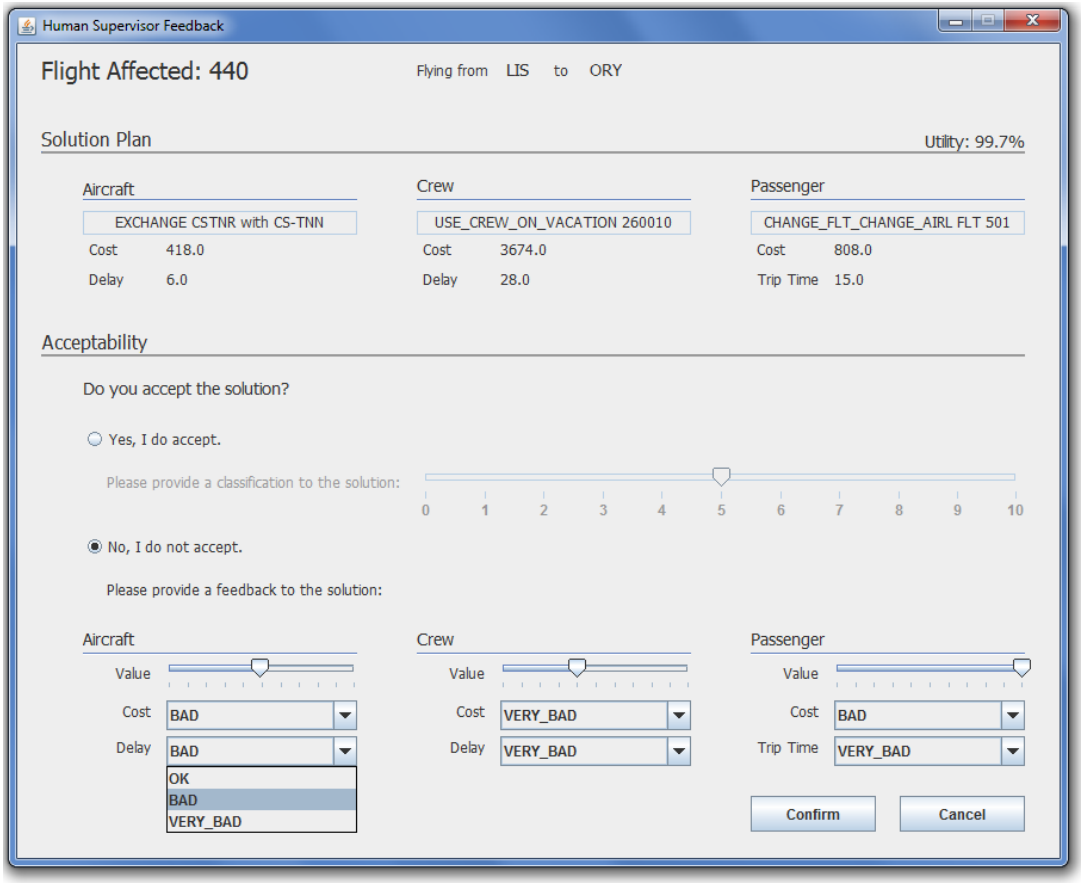

Figura 5.1: Interface *Human-in-the-Loop* - O operador humano poderá aceitar ou rejeitar a solução, tendo, em ambos os casos, que atribuir uma avaliação à solução automática.

ser gerados. Após o processo de obtenção de solução, o operador humano pode conhecer, através da mesma interface, qual a solução encontrada pelo sistema e um conjunto de parâmetros associados à avaliação dessa solução usados no processo de decisão entre os agentes. Acredita-se que esta informação permite, ao operador humano, avaliar a qualidade da solução encontrada e as implicações desta no plano operacional.

Para o desenvolvimento da funcionalidade descrita neste capítulo, foi adicionada, à interface já existente, uma outra interface que permite ao operador humano aceitar ou rejeitar a solução encontrada. Esta nova interface, apresentada na figura 5.1, é acessível a partir de um botão presente na interface inicial assim que o sistema devolva uma solução para o problema. De forma a garantir a usabilidade da funcionalidade *Human-in-the-Loop*, nesta nova interface podem ser encontradas informações sobre a solução encontrada (como sendo o número do voo afectado, a sua origem e destino, o plano para cada uma das perspectivas, os custos e atrasos associados ao plano das diferentes perspectivas) e ainda a utilidade global da solução. A reunião de todas estas informações na mesma interface onde o operador humano tem de efectuar a avaliação da solução torna-lhe a tarefa mais cómoda. Para aceitar ou rejeitar a solução, existe, na interface, um bloco composto por dois botões de opção exclusiva.

Para aceitar a solução, o humano deve seleccionar a opção "*Yes, I do accept.*"e especificar, através de um componente do tipo *slider* cuja escala varia entre zero e dez, uma avaliação global da qualidade da solução. Ao seleccionar o botão "*Confirm*", a solução e a avaliação atribuída serão guardadas numa base de dados. Conceptualmente, a aceitação da solução implica a sua utilização no contexto real. Contudo, tal consequência encontra-se fora do âmbito desta dissertação. O valor quantitativo atribuído é, no entanto, utilizado na implementação descrita neste capítulo, tópico que será referido adiante na secção 5.2.2.

Se o operador humano não ficar satisfeito com a solução deve seleccionar a opção "*No, I do not accept.*". Neste caso terá que comentar a solução, fornecendo uma avaliação mais completa. Será necessário atribuir um valor quantitativo a cada perspectiva (entre zero e dez) que traduz a importância dada pelo operador humano a essa perspectiva. Para tal, deverá utilizar os componentes do tipo *slider* denominados "*Value*", associados a cada perspectiva. Terá ainda de especificar uma valor qualitativo (*Ok*, *Bad* ou *Very Bad*) por cada atributo que compõe a solução, valor esse que deve corresponder ao grau de satisfação do operador humano face ao valor quantitativo do atributo em si. Essa classificação pode ser atribuída através dos componentes do tipo *combo box* disponíveis na interface. Ao rejeitar a solução, e seleccionando no botão "*Confirm*", o sistema iniciará um novo processo de negociação para resolução do problema. A nova solução é depois apresentada ao operador humano, que terá de voltar a avaliar, repetindo-se este processo.

A renegociação de um problema implica a alteração da forma como as propostas dos agentes *Manager* são avaliadas pelo agente *Supervisor*. É a partir da avaliação do operador humano que o sistema se adapta, alterando a forma de avaliação de propostas. Espera-se que as soluções obtidas pelo sistema, após esta alteração, sejam mais adequadas às necessidades do operador humano e, assim, ao contexto real.

A adaptação do sistema será descrita no sub-capítulo que se segue. No sub-capítulo 5.3 sumarizam-se as ideias mais relevantes deste capítulo.

5.2 Adaptação do Sistema

A interface gráfica apresentada ao operador humano é da responsabilidade do agente *Visualiser*[2], pelo que todas as informações necessárias à construção dessa interface são recolhidas por esse mesmo agente. De igual forma, quando o operador humano avalia uma solução, é o agente *Visualiser* que recolhe a avaliação e a envia ao agente *Monitor*. De cada vez que recebe essa avaliação, o agente *Monitor* procede como quando encontra um novo problema no plano operacional: comunica ao agente *Supervisor* uma mensagem que contém o problema e todas as informações necessárias à sua resolução. Nessa mensagem está incluída a avaliação do operador humano dada à última solução devolvida pelo sistema. Este comportamento é igual quer o operador humano aceite ou rejeite a solução.

Ao receber a mensagem do agente *Monitor*, o agente *Supervisor* verifica se a solução foi aceite e analisa a classificação que lhe foi atribuída. Se a solução tiver sido aceite, o agente

[2]Ver arquitectura do sistema MASDIMA na figura 2.1.

Supervisor guarda a classificação obtida e fica à espera de uma nova mensagem. Se a solução for rejeitada, o agente *Supervisor* inicia o processo de negociação automática através do protocolo GQN[3], começando por definir o valor dos parâmetros que utiliza na avaliação das propostas a receber dos agentes *Manager*. Para avaliar as propostas de solução, o agente *Supervisor* possui uma função de utilidade U, representada na equação 2.2, e valores preferenciais relativamente aos atributos *Cost* e *Delay* de todas as perspectivas. Para facilitar a leitura deste capítulo, apresenta-se novamente aqui, na equação 5.1, a função de utilidade U.

$$\begin{aligned} U =& 1 - \left(\frac{c}{\alpha_{ac} + \alpha_{cw} + \alpha_{px}} \right) \quad U \in [0,1] \\ c =& \alpha_{ac} \left(\frac{\beta_{cost\_ac} \left(\frac{cost\_ac}{max_{cost\_ac}} \right) + \beta_{delay\_ac} \left(\frac{delay\_ac}{max_{delay\_ac}} \right)}{\beta_{cost\_ac} + \beta_{delay\_ac}} \right) \\ &+ \alpha_{cw} \left(\frac{\beta_{cost\_cw} \left(\frac{cost\_cw}{max_{cost\_cw}} \right) + \beta_{delay\_cw} \left(\frac{delay\_cw}{max_{delay\_cw}} \right)}{\beta_{cost\_cw} + \beta_{delay\_cw}} \right) \\ &+ \alpha_{px} \left(\frac{\beta_{cost\_px} \left(\frac{cost\_px}{max_{cost\_px}} \right) + \beta_{tripTime\_px} \left(\frac{tripTime\_px}{max_{tripTime\_px}} \right)}{\beta_{cost\_px} + \beta_{tripTime\_px}} \right) \end{aligned} \tag{5.1}$$

Os parâmetros α traduzem a importância atribuída a cada uma das perspectivas existentes, pelo que são designados de α_{ac}, α_{cw} e α_{px}, segundo digam respeito à perspectiva *Aircraft*, *Crew Member* ou *Passenger*, respectivamente. Os parâmetros β são associados aos diferentes atributos *Cost*, *Delay* ou *TripTime* (de cada perspectiva) da solução. Os parâmetros β referentes à perspectiva *Aircraft* serão designados β_{cost_ac} e β_{delay_ac}, os da perspectiva *Crew Member* por β_{cost_cw} e β_{delay_cw} e os da perspectiva *Passenger* por β_{cost_px} e $\beta_{tripTime\_px}$. O valor de todos os parâmetros α e β são valores reais que variam no intervalo de zero a um.

Os valores preferenciais do agente *Supervisor* são utilizados na comparação com os valores de cada atributo da proposta de solução e, assim, para efectuar uma avaliação qualitativa. É esta avaliação qualitativa que o agente *Supervisor* comunica aos agentes *Manager*, pelo que o comportamento destes últimos é influenciado pelos valores preferenciais. Os valores preferenciais incluem a especificação de valores máximos e preferidos.

Na primeira negociação de qualquer problema, os valores dos parâmetros α e β utilizados pelo agente *Supervisor* correspondem a valores por omissão apresentados na tabela B.1. Estes valores foram definidos após algumas entrevistas a membros do controlo operacional da TAP Portugal responsáveis pela resolução de rupturas, numa fase anterior ao início do desenvolvimento desta dissertação. A obtenção destes valores ultrapassa o domínio desta dissertação. No caso de haver necessidade de mais informação sobre o tópico deve consultar-se o trabalho de Castro [Cas13].

A estratégia utilizada para a adaptação do sistema às preferências do operador humano passa pela alteração dos parâmetros α e β. A sua alteração tem por base a avaliação que o operador

[3]Do inglês *Generic Q-Negotiation*.

humano fornece quando rejeita a solução encontrada.

A alteração dos valores preferenciais não foi contemplada nesta dissertação. A alteração destes valores implicava estudar em detalhe o impacto no desempenho dos agentes *Manager*, pois os seus mecanismos de aprendizagem dependem desta informação. Ao alterar a forma de avaliação do agente *Supervisor*, a aprendizagem dos agentes *Manager* até então realizada poderia ficar arruinada. Para não comprometer o trabalho já realizado, optou-se por não alterar estes valores.

Os parâmetros α e β utilizados na avaliação de uma proposta de solução dependem do problema em resolução, podendo variar de problema para problema. O agente *Supervisor* mantém um registo dos últimos valores dos parâmetros utilizados para cada um dos problemas que já foram negociados. Quando é necessário fazer alterações aos parâmetros, devido à avaliação do operador humano, são sempre feitas em relação aos valores utilizados na última negociação desse problema.

Na próxima secção é descrito o processo de alteração dos parâmetros α. O processo de alteração dos parâmetros β é descrito na secção 5.2.2.

5.2.1 Alteração dos parâmetros α

Os três parâmetros α existentes na função de utilidade U do agente *Supervisor*, α_{ac}, α_{cw} e α_{px}, traduzem a importância dada a cada perspectiva do problema. Na interface de avaliação da solução encontrada (figura 5.1) existem três componentes do tipo *slider* identificado pelo título "*Value*", com uma escala quantitativa que varia de zero a dez e que traduzem a importância dada, pelo operador humano, a cada perspectiva. O valor indicado pelos componentes *slider* pode ser alterado pelo operador humano quando este escolhe rejeitar a solução encontrada. A alteração dos parâmetros α depende apenas do valor atribuído a estes componentes, ou seja, não é utilizado qualquer mecanismo de aprendizagem para determinar os valores dos parâmetros α que mais se adequam às preferências do operador humano.

Todavia, o valor atribuído pelo operador humano a cada componente *slider* não corresponde directamente ao valor atribuído a cada parâmetro α. De forma a estabelecer uma relação entre o valor dos três parâmetros, cada α depende da importância atribuída aos três componentes *slider*. O cálculo de cada parâmetro α pode ser encontrado nas equações 5.2, 5.3 e 5.4. O valor de cada componente *slider*, atribuído pelo operador humano, está identificado por $slider_{ac}$, $slider_{cw}$ e $slider_{px}$, consoante diga respeito à perspectiva *Aircraft*, *Crew Member* e *Passenger*, respectivamente.

$$\alpha_{ac} = \frac{slider_{ac}}{slider_{ac} + slider_{cw} + slider_{px}} \tag{5.2}$$

$$\alpha_{cw} = \frac{slider_{cw}}{slider_{ac} + slider_{cw} + slider_{px}} \tag{5.3}$$

$$\alpha_{px} = \frac{slider_{px}}{slider_{ac} + slider_{cw} + slider_{px}} \tag{5.4}$$

Ao decidir como calcular o valor dos parâmetros α, surgiu a seguinte questão: faz sentido variar os valores de cada parâmetro α com base no valor anterior de cada parâmetro?

Considere-se o cenário onde o operador humano rejeita a solução de um mesmo problema duas vezes. Em ambas as avaliações atribuídas, o operador humano especificou diferentes valores para as importâncias de cada perspectiva. Por um lado, e se considerarmos que existe apenas um operador humano a utilizar o sistema, a importância dada às diferentes perspectivas de um problema não deveria alterar significativamente. Contudo, dada a existência de turnos entre os membros do controlo operacional, pode acontecer que o operador humano a utilizar o sistema não seja necessariamente o mesmo. Logo, as preferências podem alterar-se. Por outro lado, o contexto real do problema pode mudar subitamente. Mesmo existindo apenas um operador humano, este pode ter a necessidade de atribuir radicalmente mais (ou menos) importância a uma perspectiva. Se os valores actuais dependessem dos anteriores, seria necessário mais interacções entre o operador humano e o sistema, tornando o processo ineficiente. Por estas razões, optou-se por fazer variar os parâmetros α independentemente do seu valor nas negociações anteriores.

5.2.2 Alteração dos parâmetros β

Os seis parâmetros β existentes na função de utilidade U do agente *Supervisor*, β_{cost_ac}, β_{delay_ac}, β_{cost_cw}, β_{delay_cw}, β_{cost_px} e $\beta_{tripTime\_px}$ traduzem a importância dada a cada atributo de cada perspectiva do problema.

A alteração destes parâmetros é da responsabilidade de um mecanismo de aprendizagem por reforço implementado no agente *Supervisor*. Os conceitos de aprendizagem concorrente ou em equipa não se aplicam neste domínio, pois esse mecanismo é o único ao nível do agente *Supervisor*. O facto do mecanismo de aprendizagem do agentes *Supervisor* poder influenciar os mecanismos de aprendizagem implementados nos agentes *Manager* constitui outra questão, que só pode ser respondida através da experiência. A interferência entre os mecanismos de aprendizagem dos agentes *Manager* e do agente *Supervisor* será abordada no capítulo 6.

A função deste mecanismo de aprendizagem, constituído por apenas um motor, é permitir que o agente *Supervisor*, que representa o sistema, aprenda as preferências do operador humano em relação aos atributos da solução e os avalie de acordo com as necessidades do contexto real. A informação de que o agente *Supervisor* dispõe consiste na avaliação qualitativa que é atribuída pelo operador humano (através da interface representada na figura 5.1) a cada atributo da solução encontrada pelo sistema.

A razão pela qual a avaliação dada pelo operador humano é qualitativa reside no facto de se considerar o operador humano ineficiente na avaliação de todas as restrições existentes e, por conseguinte, na procura da solução óptima. Assim, não fará sentido que o operador humano especifique um valor exacto para os atributos que representam os custos e atrasos de um plano. Poderia especificar valores máximos ou preferidos, que não deveriam ser ultrapassados, mas estes valores já são utilizados e correspondem aos valores preferidos do agente *Supervisor*.

O operador humano deve avaliar cada atributo com um de três valores qualitativos possíveis para avaliar cada atributo: *Good*, *Bad* e *VeryBad*. Se considerar o valor quantitativo do atributo suficientemente bom para ser aceite, o operador humano deverá atribuir a classificação *Good*. No caso de o operador humano querer ver o valor melhorado, embora não seja um valor que lhe

desagrade em demasia, deve classificá-lo como *Bad*. No caso do valor quantitativo for muito aquém das suas expectativas, o operador humano deve atribuir a classificação *VeryBad*.

5.2.2.1 Definição de estado, acção e recompensa

O funcionamento do motor de aprendizagem do agente *Supervisor* assemelha-se, em alguns aspectos, ao funcionamento dos motores dos mecanismos de aprendizagem apresentados no capítulo 4. É com base na avaliação atribuída pelo operador humano que o motor de aprendizagem do agente *Supervisor* funciona. Essa avaliação é representada no estado segundo o n-tuplo da expressão 5.5, em conjunto com o problema.

$$e =< cause, class_{Cost\_ac}, class_{Delay\_ac}, class_{Cost\_cw}, class_{Delay\_cw}, class_{Cost\_px}, class_{TripTime\_px} > \tag{5.5}$$

As razões que justificam a representação do estado são semelhantes às já identificadas no capítulo 4: é necessário que o agente consiga perceber qual é o problema em negociação e qual é a sua situação perante o operador humano. O estado deve apresentar informações sobre o problema em questão e a avaliação recebida. O elemento *cause* indica qual foi a causa que originou o problema. Os treze valores possíveis deste elemento estão descritos na tabela A.1. Os elementos $class_{Cost\_ac}$, $class_{Delay\_ac}$, $class_{Cost\_cw}$, $class_{Delay\_cw}$, $class_{Cost\_px}$ e $class_{TripTime\_px}$ traduzem a avaliação qualitativa dada pelo operador humano a cada atributo que compõe a solução. A inclusão dos seis elementos é essencial, pois o agente *Supervisor* necessita de fazer uma análise global à proposta.

Como já foi referido no capítulo 3, o tamanho do espaço de estados influencia a convergência do algoritmo. Quanto maior for o espaço de estados mais lenta é a convergência do algoritmo. Na representação do estado do mecanismo descrito no capítulo 4 existiam dois elementos que permitiam caracterizar o problema: *cause* e *resource*. Os valores possíveis deste último diferem consoante a perspectiva do problema. Neste mecanismo de aprendizagem não existe esta divisão por perspectivas. Ainda que não possua conhecimento que lhe permita resolver o problema, o agente *Supervisor* é responsável por fazer uma análise global da solução. Ao incluir informação referente ao recurso na representação do estado, no mecanismo do agente *Supervisor*, seria necessário incluir um elemento por cada recurso de cada perspectiva. Tendo em conta que existem cinco recursos da perspectiva *Aircraft* e mais dois de cada uma das perspectivas restantes, a inclusão destes três elementos iria ter um grande impacto na convergência do algoritmo. Embora a inclusão deste elemento permitisse uma melhor caracterização do problema, pensa-se que as vantagens não superam o problema de convergência. Assim, poderá existir um espaço de estados com, no máximo, 9477[4] estados.

A representação da acção corresponde ao n-tuplo da expressão 5.6. Cada elemento representa a acção a tomar em relação ao parâmetro β correspondente. Os valores destes elementos, bem

[4]Valor obtido a partir da multiplicação da quantidade de valores possíveis para cada elemento que constitui um estado. O elemento *cause* pode assumir 13 valores, enquanto que os restantes seis elementos podem assumir 3 valores distintos: $13 \times 3 \times 3 \times 3 \times 3 \times 3 = 9477$ estados.

como o seu significado, podem ser consultados na tabela A.5 do anexo A.

$$a =< action_{Cost\_ac}, action_{Delay\_ac}, action_{Cost\_cw}, action_{Delay\_cw}, action_{Cost\_px}, action_{TripTime\_px} > \tag{5.6}$$

Considerando que cada elemento que compõe a acção pode assumir três valores, cada estado poderá ter, no máximo, 729^5 acções.

Na primeira execução do algoritmo, as tabelas dos valores Q de cada par estado-acção são iniciadas com o valor 0.0. A actualização do valor Q de cada par estado-ação depende do algoritmo em utilização ($Q-Learning$ ou *Sarsa*) e é feita segundo as equações 3.1 ou 3.2, onde R é a função de recompensa definida na equação 5.7 e pen_i, definida na equação 5.8, é a penalização associada ao valor obtido de classificação do atributo. No cálculo da recompensa R o valor da constante m é 6, visto que o estado contempla a classificação dos seis atributos da proposta de solução.

$$R = \begin{cases} m & \text{se aceite,} \\ \frac{m}{2} - \sum_{i=1}^{m} pen_i & \text{se não aceite.} \end{cases} \qquad m \text{ - número de atributos utilizados} \tag{5.7}$$

$$pen_i = \begin{cases} 0.8 & \text{se Very Bad,} \\ 0.4 & \text{se Bad,} \\ 0.0 & \text{se Good.} \end{cases} \tag{5.8}$$

5.2.2.2 Processo de Aprendizagem

Depois de definido o conceito de estado, acção e recompensa, interessa agora descrever o ciclo do processo de aprendizagem. Note-se que, na primeira negociação o operador humano não pode especificar qualquer tipo de avaliação, pelo que os valores dos parâmetros α e β utilizados pelo agente *Supervisor* na primeira negociação correspondem aos valores por omissão da tabela B.1. A primeira negociação é processada de forma absolutamente automática, isto é, sem qualquer tipo de intervenção do operador humano. O mecanismo de aprendizagem é executado no final de cada negociação, quando o agente *Monitor* comunica a avaliação do operador humano ao agente *Supervisor*. No diagrama de sequência da figura 5.2 encontra-se graficamente representado o ciclo de aprendizagem do agente *Supervisor*.

Processo de aprendizagem do agente *Supervisor*:

1. O operador humano é confrontado com a primeira solução devolvida pelo sistema para um dado problema. Essa solução foi avaliada pelo agente *Supervisor* com os valores por omissão. O operador humano avalia a solução, podendo aceitá-la ou rejeitá-la. Essa avaliação é comunicada ao agente *Visualiser*.

2. O agente *Visualiser* envia a avaliação ao agente *Monitor*.

[5] $3 \times 3 \times 3 \times 3 \times 3 \times 3 = 729$ acções.

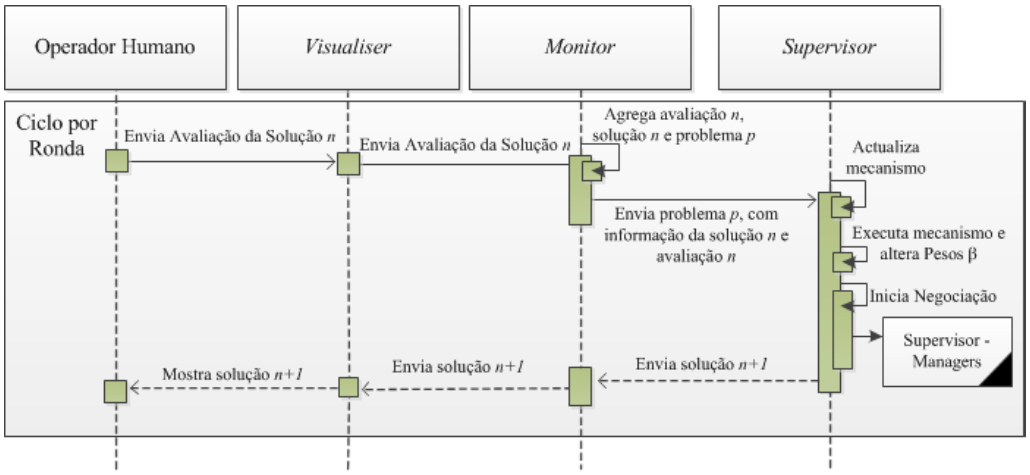

Figura 5.2: Protocolo GQN com Aprendizagem no Agente *Supervisor* - Diagrama de sequência que representa interacção do operador humano com o sistema MASDIMA e o processo de aprendizagem do mecanismo do agente *Supervisor*.

3. O agente *Monitor* agrega a avaliação recebida à solução e problema respectivos. Envia estas informações ao agente *Supervisor* seguindo o protocolo GQN.

4. O agente *Supervisor* analisa a informação recebida e actualiza o seu mecanismo de aprendizagem. De seguida, procede do seguinte modo:

   - Se a solução tiver sido aceite pelo operador humano, termina este processo e espera por nova comunicação da parte do agente *Monitor*.
   - Se a solução tiver sido rejeitada, inicia o processo de alteração dos valores dos parâmetros α e β. Aquando dos parâmetros β, selecciona uma nova acção através da execução do algoritmo de aprendizagem e altera os valores dos parâmetros utilizados na última negociação de acordo com a acção seleccionada.

5. Após a alteração dos valores de todos os parâmetros, o agente *Supervisor* inicia a negociação automática com os agentes *Manager*.

6. Quando a negociação termina, o agente *Supervisor* envia a solução vencedora ao agente *Monitor*.

7. O agente *Monitor* guarda a solução recebida e comunica-a ao agente *Visualiser*.

8. O agente *Visualiser* disponibiliza as informações necessárias para a avaliação da solução ao operador humano através da interface gráfica.

Como descrito no processo de aprendizagem, o agente *Supervisor* tem de adaptar os valores dos parâmetros β à acção seleccionada pelo mecanismo de aprendizagem.

Os valores β são alterados segundo a acção determinada para cada parâmetro. Tome-se como exemplo a acção representada no n-tuplo da expressão 5.9.

$$a =< Inc, Dec, Keep, Inc, Keep, Keep > \tag{5.9}$$

Segundo a acção seleccionada, o valor dos parâmetros β_{Cost_ac} e β_{Delay_cw} deve aumentar, os valores dos parâmetros β_{Cost_cw}, β_{Cost_px} e $\beta_{TripTime\_px}$ devem permanecer os mesmos e, por fim, o valor do parâmetro β_{Delay_ac} deve diminuir. Os parâmetros β são alterados segundo a equação 5.10.

$$\beta_n = \begin{cases} \beta_{n-1} * (1+z) & \text{se Inc,} \\ \beta_{n-1} & \text{se Keep,} \\ \beta_{n-1} * (1-z) & \text{se Good.} \end{cases} \tag{5.10}$$

β_n representa um dos parâmetros β que vai ser utilizado na avaliação de propostas da próxima negociação n. β_{n-1} representa o valor do mesmo parâmetro β na negociação anterior a n. O agente *Supervisor* altera os valores dos diferentes parâmetros β em função dos seus valores imediatamente anteriores. z é uma constante com valor igual a 0.05 que permite alterar o anterior valor do parâmetro em pequenas proporções. Este valor foi determinado de forma empírica.

5.2.2.3 Algoritmo de Aprendizagem

Neste trabalho apenas se testou o mecanismo de aprendizagem descrito na secção anterior com o algoritmo *Q-Learning*, embora seja possível a sua utilização com o algoritmo *Sarsa*. A razão de não se ter testado o mecanismo utilizando o algoritmo *Sarsa* reside no facto de as experiências preliminares com os mecanismos no contexto da negociação automática entre agentes (descrito no capítulo 4) não terem revelado grandes diferenças entre os desempenhos de ambos os algoritmos. A estratégia de selecção de acções utilizada foi também a estratégia $Softmax$. O parâmetro t, utilizado pela estratégia *Softmax* na distribuição de *Boltzmann* (equação 3.3), varia segundo a equação 5.11.

$$t = t - 0.9^n, \quad t_{inicial} = 12, \quad n\text{- número de iterações} \tag{5.11}$$

O valor inicial do parâmetro α da equação 3.1 corresponde a 0.9. Esse mesmo valor é actualizado segundo a equação 5.12.

$$\alpha = \frac{\alpha_{inicial}}{1 + \beta * n}, \quad n \text{ - número de iterações} \tag{5.12}$$

A constante β da equação 5.12 tem o valor de 0.05 e permite fazer decrescer mais lentamente, em cada iteração n, o valor de α. O parâmetro γ da equação 3.1 tem o valor de 0.6.

Como já tinha sido referido no capítulo 4, os valores utilizados no algoritmo têm por base o estudo de Rocha [Roc01]. Durante as experiências preliminares com o mecanismo de aprendizagem do agente *Supervisor*, os valores mantiveram-se aptos ao contexto, pelo que foram preservados.

5.3 Resumo

Neste capítulo foi apresentada a solução desenvolvida para o problema identificado no subcapítulo 2.4. O objectivo é permitir ao operador humano validar as soluções automáticas obtidas pelo sistema. Esta funcionalidade permite tornar o sistema mais facilmente adaptável a situações pouco comuns. É ainda com o objectivo de o tornar socialmente aceitável que se inclui esta funcionalidade no sistema MASDIMA.

Quer aceite ou rejeite a proposta, o operador humano tem de especificar uma avaliação. No caso da solução ser rejeitada, a negociação automática entre os agentes deve ser reiniciada.

A solução desenvolvida passa por alterar a função de utilidade que o agente *Supervisor* utiliza na negociação automática para a avaliação das propostas de solução dos agentes *Manager*. A função de utilidade possui dois tipos de parâmetros que representam a importância atribuída a elementos da solução: α e β. A alteração da função de utilidade é conseguida através da alteração do valor destes parâmetros.

Os parâmetros α traduzem a importância atribuída a cada perspectiva do problema. Na solução desenvolvida, estes parâmetros são alterados segundo a importância quantitativa especificada pelo operador humano na avaliação da última proposta de solução para o problema. Os parâmetros β traduzem a importância de cada atributo que compõe a proposta de solução. Já estes parâmetros, na solução desenvolvida, são alterados com recurso a um processo de aprendizagem por reforço. Muitos dos conceitos apreendidos durante o desenvolvimento da solução para o problema da aprendizagem na negociação automática, descrito no capítulo 4, foram também aplicados no desenvolvimento deste processo de aprendizagem.

Capítulo 6

Experiências

Neste capítulo são apresentadas as experiências efectuadas sobre as soluções desenvolvidas no âmbito desta dissertação, cuja descrição pode ser encontrada nos capítulos 4 e 5.

6.1 Introdução

À semelhança dos capítulos anteriores, também neste existe uma separação entre os dois tópicos já identificados neste trabalho: aprendizagem na negociação automática e interacção de um operador humano com o sistema MASDIMA[1]. Todavia, os dados, as abordagens e as métricas utilizadas nas experiências são partilhados por ambos os tópicos. A definição dos dados, das abordagens e das métricas utilizadas na experimentação será efectuada antes das experiências, nos sub-capítulos 6.2, 6.3 e 6.4. No sub-capítulo 6.5 analisam-se os resultados das experiências efectuadas com o mecanismo de aprendizagem na negociação automática (primeiro tópico). Finalmente, no sub-capítulo 6.6, são descritas as experiências relacionadas com o tópico *Human-in-the-Loop* (segundo tópico). No sub-capítulo 6.7 são resumidas as conclusões mais importantes deste capítulo.

6.2 Dados Utilizados

O conjunto de dados utilizado durante o desenvolvimento desta dissertação foi disponibilizado pela companhia aérea TAP Portugal. Correspondem a dados reais obtidos num mês de operações (Setembro de 2009) realizadas na TAP Portugal. O conjunto de dados utilizados neste trabalho não corresponde à versão original entregue pela TAP Portugal. A quantidade e complexidade dos dados desta companhia aérea exigiram um tratamento que permitisse filtrar apenas a informação necessária para o protótipo desenvolvido. Ainda assim, a informação utilizada é bastante completa

[1] Do inglês *Multi-Agent System for Disruption Management*.

e possui detalhes não só sobre os voos e problemas associados mas também sobre os recursos de todas as perspectivas (avião, tripulação e passageiros).

Estes dados já haviam sido utilizados para testar o Protótipo Base do MASDIMA, e até versões anteriores [Cas13]. Decidiu-se, por isso, usar o mesmo conjunto de dados por forma a ser possível comparar as abordagens desenvolvidas no âmbito desta dissertação e as abordagens anteriores.

Na tabela C.1 do anexo C, apresentam-se as informações extraídas do conjunto de dados disponibilizados pela TAP Portugal [Cas13]. Desses dados é possível obter informações sobre vários custos relacionados com as três perspectivas. Associada à perspectiva do avião existe, por exemplo, informação sobre as taxas cobradas por cada aeroporto para os diferentes aviões, custos de combustível e custos de manutenção. Associada à perspectiva da tripulação existe informação desde os salários associados a cada membro, até taxas associadas a dias extras ou ainda custos relacionados com hotéis. Associada à perspectiva dos passageiros existe informação quanto ao custo de compensação, no caso de existirem atrasos ou cancelamentos de voos, e até um valor denominado *goodwill* que pretende simular as perdas da companhia aérea através da satisfação do cliente. O cálculo dos custos de uma solução, embora não seja descrito nesta dissertação, tem por base este tipo de informação.

Na tabela C.2 está caracterizado o plano operacional referente ao mês de Setembro de 2009. Segundo Castro [Cas13], os dados deste mês apresentam características bastante similares à média de um ano de operações, razão pela qual foram utilizados apenas os dados desse mês.

Na tabela C.3 encontram-se caracterizados os problemas originados por rupturas no plano operacional da tabela C.2. São considerados 49 problemas distintos, que afectam um total de 31 aviões, 286 membros de tripulação e 4760 passageiros. Os custos totais representados na tabela C.3 referem-se ao custo de efectuar as operações com os atrasos inerentes, sem a alteração do plano operacional. Note-se que não são referidos custos para a perspectiva dos passageiros. Tal facto deve-se à assumpção que, se não houver alterações no plano operacional inicial, a companhia aérea não tem prejuízos nesta perspectiva. Esses prejuízos apenas se verificam se a companhia aérea alterar o seu plano operacional, caso em que terá que realocar os seus passageiros.

6.3 Abordagens

Por abordagem entende-se, no âmbito deste trabalho, o método utilizado para gerir as rupturas do plano operacional. A configuração do sistema MASDIMA não é a mesma em cada abordagem, o que dá origem a diferentes métodos de resolução das mesmas rupturas. Existem seis abordagens distintas, que são descritas de seguida:

- **ST - Abordagem Sequencial Típica:** Nesta abordagem não é utilizada a negociação automática entre agentes. A resolução das rupturas é sequencial, tal como é feita pelos operadores humanos do centro de controlo operacional. Nesta resolução sequencial é considerada primeiro a perspectiva do avião, seguida da perspectiva da tripulação e terminando na perspectiva dos passageiros. Ou seja, ao resolver o problema para uma perspectiva, a solução

escolhida funciona como restrição para a resolução das próximas perspectivas. O resultado final é aplicado sobre o problema sem consideração por outras soluções possíveis.

- **NB - Abordagem Negociação Base:** Nesta abordagem é utilizado o Protótipo Base do sistema MASDIMA, cujo funcionamento foi descrito no capítulo 2. Neste protótipo não é utilizada a aprendizagem na negociação automática entre agentes. O agente *Supervisor* comunica, aos agentes *Manager*, uma avaliação qualitativa relativa aos atributos que compõem cada proposta, mas a estratégia dos agentes *Manager* é estática. Estes limitam-se a tentar adaptar pelo menos o valor de um dos atributos da proposta seguinte à última avaliação recebida, sem efectuar qualquer tipo de aprendizagem. Não existe qualquer interacção do operador humano com o sistema. O agente *Supervisor* negoceia cada problema apenas uma vez com recurso aos valores por omissão dos parâmetros α e β da sua função de avaliação (descritos no capítulo 5).

- **NQmA - Abordagem Negociação *Q-Learning* com o mecanismo *A*:** Esta abordagem equivale à solução descrita no sub-capítulo 4.2.1, onde é utilizada aprendizagem na negociação automática entre agentes através do mecanismo *A* e com o algoritmo *Q-Learning*. O mecanismo de aprendizagem *A* é composto por três motores distintos por cada agente *Manager* (ver figura 4.1). O estado e a acção do algoritmo são representados pelos n-tuplos das expressões 4.1 e 4.5, respectivamente. Não existe qualquer interacção do operador humano com o sistema. O agente *Supervisor* negoceia cada problema apenas uma vez com recurso aos valores por omissão dos parâmetros α e β da sua função de avaliação (descritos no capítulo 5).

- **NQmA-DOM - Abordagem Negociação *Q-Learning* com o mecanismo *A* (versão inicial):** Esta abordagem possui apenas uma diferença em relação à abordagem NQmA. Na parte final da descrição do processo de aprendizagem no mecanismo *A* (secção 4.2.1.2) foi descrita uma alteração à representação da acção neste mecanismo, que consistiu na remoção do elemento $action_{Domain}$. Nesta abordagem, a representação da acção segue o n-tuplo da expressão 4.2, de forma a justificar o abandono da representação com o elemento $action_{Domain}$. Tudo o resto segue a descrição de NQmA.

- **NSmA - Abordagem Negociação *Sarsa* com o mecanismo *A*:** Esta abordagem corresponde à solução descrita no sub-capítulo 4.2.1, onde é utilizada aprendizagem na negociação automática entre agentes através do mecanismo *A* e com o algoritmo *Sarsa*. O mecanismo de aprendizagem *A* é composto por três motores distintos por cada agente *Manager* (ver figura 4.1). O estado e a acção do algoritmo são representados pelas fórmulas 4.1 e 4.5, respectivamente. Não existe qualquer interacção do operador humano com o sistema. O agente *Supervisor* negoceia cada problema apenas uma vez com recurso aos valores por omissão dos parâmetros α e β (tabela B.1 do anexo B) da sua função de avaliação.

- **NQmB - Abordagem Negociação *Q-Learning* com o mecanismo *B*:** Esta abordagem equivale à solução descrita no sub-capítulo 4.2.2, onde é utilizada aprendizagem na negociação automática entre agentes através do mecanismo *B* e com o algoritmo *Q-Learning*. O mecanismo de aprendizagem *B* é composto por um motor de aprendizagem por cada agente *Manager* (ver figura 4.3). Não existe qualquer interacção do operador humano com o sistema. O agente *Supervisor* negoceia cada problema apenas uma vez com recurso aos valores por omissão dos parâmetros α e β (tabela B.1 do anexo B) da sua função de avaliação.

- **NSmB - Abordagem Negociação *Sarsa* com o mecanismo B:** Esta abordagem equivale à solução descrita no sub-capítulo 4.2.2, onde é utilizada aprendizagem na negociação automática entre agentes através do mecanismo *B* e com o algoritmo *Sarsa*. A aprendizagem é efectuada pelos agentes *Manager*, que devem aprender a efectuar propostas que melhor se adequem às preferências do agente *Supervisor* e, dessa forma, vencer a negociação. O mecanismo de aprendizagem *B* é composto por um motor de aprendizagem por cada agente *Manager* (ver figura 4.3). Não existe qualquer interacção do operador humano com o sistema. O agente *Supervisor* negoceia cada problema apenas uma vez com recurso aos valores por omissão dos parâmetros α e β (tabela B.1 do anexo B) da sua função de avaliação.

- **NH - Abordagem Negociação com *Human-in-the-Loop*:** Esta abordagem equivale à solução descrita no capítulo 5. Nesta solução o operador humano pode interagir com o sistema através da avaliação da solução devolvida pelo sistema, no fim da negociação. Essa avaliação permite alterar os valores por omissão dos parâmetros α e β da sua função de avaliação, através das estratégias descritas no sub-capítulo 5.2. Nesta abordagem mantém-se a aprendizagem na negociação automática entre agentes, sendo que as configurações desta funcionalidade são as mesmas que na abordagem NQmB: é utilizada aprendizagem na negociação automática entre os agentes através da utilização do mecanismo *B* com o algoritmo *Q-Learning*.

Cada abordagem, com excepção da abordagem NH, foi testada com 100 execuções do sistema. Uma execução *e* do sistema envolve a resolução de 49 problemas existentes.

Na abordagem NH, limitou-se o número total de execuções do sistema a 25, por existir a necessidade de colocar um operador humano a validar as soluções devolvidas pelo sistema MASDIMA. As experiências foram realizadas com apenas um operador humano. Nesta abordagem, o número total de negociações está dependente da aceitação das soluções por parte do operador humano. Ao rejeitar uma solução, o operador humano está a forçar o sistema a efectuar uma nova negociação. Uma execução do sistema só terminará quando o operador humano aceitar uma solução por problema existente. No mínimo, existirão sempre 49 negociações de problemas por execução *e* do sistema (uma negociação por cada problema). Contudo, o total de negociações por uma execução do sistema pode ser superior a 49.

6.4 Métricas

Algumas das métricas apresentadas neste capítulo já haviam sido utilizadas por Castro [Cas13] em experiências de protótipos anteriores. São utilizadas nestas experiências de forma a haver termo de comparação com as novas abordagens agora desenvolvidas. Outras métricas, no entanto, são novas e foram criadas para avaliar o desempenho do sistema após o desenvolvimento das funcionalidades descritas neste documento. As métricas criadas no âmbito desta dissertação são sete: **Diferença Média de Utilidades**, **Rácio Médio de Recuperação de Atrasos à Partida de Voos**, **Rácio Médio de Recuperação de Custos de Voos**, **Rácio Médio de Recuperação de Custos da Tripulação**, **Número de Negociações Até Aceitação**, **Qualidade Média Atribuída pelo Operador Humano** e **Satisfação Média do Operador Humano**.

As métricas já existentes e consideradas nas experiências realizadas são descritas nos pontos seguintes:

- **M1 - Utilidade Média Global do Sistema:** Valor médio da utilidade global das soluções vencedoras para o agente *Supervisor*. Pode ter um valor no intervalo [0,1], sendo que valores mais elevados são melhores.

$$\overline{U_{global}} = \frac{\sum_{e=1}^{n}(\overline{U_{global}}_e)}{n} \tag{6.1}$$

  $\overline{U_{global}}_e$ é a média aritmética das utilidades globais das soluções vencedoras para os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M2 - Utilidade Média do agente *Aircraft Manager*:** Valor médio da utilidade individual das soluções vencedoras para o agente *Aircraft Manager*. Pode ter um valor no intervalo [0,1], sendo que valores mais elevados são melhores.

$$\overline{U_{ac}} = \frac{\sum_{e=1}^{n}(\overline{U_{ac}}_e)}{n} \tag{6.2}$$

  $\overline{U_{ac}}_e$ é a média aritmética das utilidades individuais, para o agente *Aircraft Manager*, das soluções vencedoras para os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M3 - Utilidade Média do agente *Crew Member Manager*:** Valor médio da utilidade individual das soluções vencedoras para o agente *Crew Member Manager*. Pode ter um valor no intervalo [0,1], sendo que valores mais elevados são melhores.

$$\overline{U_{cw}} = \frac{\sum_{e=1}^{n}(\overline{U_{cw}}_e)}{n} \tag{6.3}$$

  $\overline{U_{cw}}_e$ é a média aritmética das utilidades individuais, para o agente *Crew Member Manager*, de cada solução vencedora para todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M4 - Utilidade Média do agente *Passenger Manager*:** Valor médio da utilidade individual das soluções vencedoras para o agente *Passenger Manager*. Pode ter um valor no intervalo [0,1], sendo que valores mais elevados são melhores.

$$\overline{U_{px}} = \frac{\sum_{e=1}^{n}(\overline{U_{px}}_e)}{n} \tag{6.4}$$

  $\overline{U_{px}}_e$ é a média aritmética das utilidades individuais, para o agente *Passenger Manager*, de cada solução vencedora para todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M5 - Bem-estar Social Médio:** Esta métrica tem por base a definição de Sandholm [San99], que define o bem-estar social como sendo a soma das utilidades individuais dos agentes do sistema. Neste contexto específico, o bem-estar social médio $\overline{U_{bs}}$ reflecte a utilidade média individual de todos os agentes *Manager*. Quanto mais elevado for o seu valor melhor. É calculado através da soma da utilidade média individual dos três agentes *Manager*, apresentadas nas equações 6.2, 6.3 e 6.4:

$$\overline{U_{bs}} = \overline{U_{ac}} + \overline{U_{cw}} + \overline{U_{px}} \tag{6.5}$$

- **M6 - Δ Médio para a Solução Óptima:** Considerando o plano operacional inicial como solução óptima, compara-se a diferença entre a utilidade da solução óptima ($Uopt_i$) e a utilidade da solução devolvida pelo sistema ($Uglobal_i$), para cada problema i. Através deste valor é possível comparar a solução obtida com a solução do plano inicial. Valores menores correspondem a soluções melhores, visto que a utilidade de se executar uma ou outra solução é próxima. No caso do valor ser negativo, considera-se que a solução é mais útil que a solução inicial.

$$\overline{\Delta(optimal)}_i = Uopt_i - Uglobal_i \tag{6.6}$$

$$\overline{\Delta(optimal)} = \frac{\sum_{e=1}^{n}(\overline{\Delta(optimal)}_e)}{n} \tag{6.7}$$

  $\overline{\Delta(optimal)}_e$ é a média aritmética da diferença entre as utilidades das soluções óptimas e as utilidades das soluções devolvidas pelo sistema, para todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M7 - Atraso Médio à Partida de um Voo:** Corresponde à média de minutos de atraso de partida do voo (*flight delay*) inerentes às soluções automáticas devolvidas pelo sistema. Quanto menores forem os atrasos melhor.

$$FD_i = etd_i - etd_i \tag{6.8}$$

$$\overline{FD} = \frac{\sum_{e=1}^{n}(\overline{FD}_e)}{n} \tag{6.9}$$

FD_i é o atraso à partida de um voo para o problema i. etd_i é a nova hora de partida do voo (que inclui atrasos) e std_i é a hora de partida do voo no plano inicial. $\overline{FD}_e$ é a média aritmética do atraso à partida dos voos de todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M8 - Atraso Médio da Tripulação:** Corresponde à média de minutos de atraso dos membros da tripulação (*crew member delay*) inerentes às soluções automáticas devolvidas pelo sistema. Quanto menor for este valor, melhor.

$$CD_i = esign_i - ssign_i \tag{6.10}$$

$$\overline{CD} = \frac{\sum_{e=1}^{n}(\overline{CD}_e)}{n} \tag{6.11}$$

CD_i é o atraso na entrada ao serviço (*sign on time*) dos membros da tripulação para o problema i. $esign_i$ é a nova hora de entrada ao serviço (que inclui atrasos) e $ssign_i$ é a hora de entrada ao serviço no plano inicial. $\overline{CD}_e$ é a média aritmética do atraso da tripulação dos voos de todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M9 - Atraso Médio no Tempo de Viagem dos Passageiros:** Corresponde à média de minutos de atraso no tempo de viagem de todos os passageiros (*passenger delay*) inerentes às soluções automáticas devolvidas pelo sistema. Quanto menor for este valor, melhor.

$$PD_i = ett_i - stt_i \tag{6.12}$$

$$\overline{CD} = \frac{\sum_{e=1}^{n}(\overline{PD}_e)}{n} \tag{6.13}$$

PD_i é o atraso na entrada ao serviço (*sign on time*) dos membros da tripulação para o problema i. ett_i é o tempo de viagem real para os passageiros (que inclui atrasos) e stt_i é o tempo de viagem no plano inicial. $\overline{PS}_e$ é a média aritmética do atraso no tempo de viagem de todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M10 - Custos Médios de um Voo e Avião:** Corresponde à média de custos, em unidades monetárias, relacionados com a perspectiva do avião e voo inerentes às soluções automáticas devolvidas pelo sistema. Quanto menores forem os custos melhor.

$$\overline{FC} = \frac{\sum_{e=1}^{n}(\overline{FC}_e)}{n} \tag{6.14}$$

$\overline{FC_e}$ é a média aritmética dos custos dos voos e avião de todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M11 - Custos Médios da Tripulação:** Corresponde à média de custos, em unidades monetárias, relacionados com a perspectiva da tripulação inerentes às soluções automáticas devolvidas pelo sistema. Quanto menores forem os custos melhor.

$$\overline{CC} = \frac{\sum_{e=1}^{n}(\overline{CC}_e)}{n} \tag{6.15}$$

$\overline{CC}_e$ é a média aritmética dos custos da tripulação de todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M12 - Custos Médios dos Passageiros:** Corresponde à média de custos, em unidades monetárias, relacionados com a perspectiva dos passageiros inerentes às soluções automáticas devolvidas pelo sistema. Quanto menores forem os custos melhor.

$$\overline{PC} = \frac{\sum_{e=1}^{n}(\overline{PC}_e)}{n} \tag{6.16}$$

$\overline{PC}_e$ é a média aritmética dos custos dos passageiros de todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M13 - Número Médio de Rondas:** Corresponde à média de número de rondas necessárias, na negociação automática entre agentes, para chegar à solução final do problema. Quanto menor for o número de rondas melhor.

$$\overline{NR} = \frac{\sum_{e=1}^{n}(\overline{NR}_e)}{n} \tag{6.17}$$

$\overline{NR}_e$ é a média aritmética do número de rondas necessárias para chegar à solução de todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

As novas métricas criadas no âmbito do trabalho desta dissertação são as seguintes:

- **M14 - Diferença Média de Utilidades:** Para a mesma abordagem, são consideradas, entre as utilidades médias individuais dos três agentes *Manager*, aquelas que têm maior e menor valor. A diferença entre estas utilidades constitui a diferença $\overline{Dif_U}$ de utilidades para uma abordagem. Com esta métrica pretende-se verificar se existe um equilíbrio entre a utilidade dos agentes *Manager*, já que o objectivo da negociação automática entre agentes passa por aumentar a utilidade individual de todos agentes que apresentam soluções. Assim, valores mais perto do zero são melhores.

$$\overline{Dif_U} = max(\overline{U_{ac}}, \overline{U_{cw}}, \overline{U_{px}}) - min(\overline{U_{ac}}, \overline{U_{cw}}, \overline{U_{px}}) \tag{6.18}$$

- **M15 - Rácio Médio de Recuperação de Atrasos à Partida de Voos:** Este rácio estabelece uma relação entre os minutos de atraso inerentes à solução automática devolvida pelo sistema para um problema *i* (FD) e os minutos de atraso à partida previstos para um voo antes da resolução desse mesmo problema (OD). Valores mais elevados são melhores. Valores perto do 1 indicam que as soluções automáticas devolvidas pelo sistema possuem menores atrasos que o problema original. Valores perto do 0 indicam que as soluções possuem atrasos próximos aos do problema original. Se o valor for negativo, os atrasos das soluções automáticas ultrapassam os atrasos iniciais.

$$FDrcv_i = \frac{FD_i}{OD_i} \tag{6.19}$$

$$\overline{FDrcv} = 1 - (\frac{\sum_{e=1}^{n}(\overline{FDrcv}_e)}{n}) \tag{6.20}$$

FD_i é calculado segundo a equação 6.8. $FDrcv_i$ é o rácio de atrasos para um problema *i*. $\overline{FDrcv}_e$ é a média aritmética dos rácios de atrasos das soluções para todos os problemas incluídos numa execução *e* do sistema e *n* é o número total de execuções.

- **M16 - Rácio Médio de Recuperação de Custos de Voos:** Este rácio estabelece uma relação entre os custos inerentes à solução automática devolvida pelo sistema para um problema *i* (FC_i) e os custos previstos para um voo antes da resolução desse mesmo problema (OC_i). Valores mais elevados são melhores. Valores perto do 1 indicam que as soluções automáticas devolvidas pelo sistema possuem menores custos que o problema original. Valores perto do 0 indicam que as soluções possuem custos próximos aos do problema original. Se o valor for negativo, os custos das soluções automáticas ultrapassam os custos iniciais.

$$FCrcv_i = \frac{FC_i}{OC_i} \tag{6.21}$$

$$\overline{FCrcv} = 1 - (\frac{\sum_{e=1}^{n}(\overline{FCrcv}_e)}{n}) \tag{6.22}$$

FC_i é o cálculo de todos os custos relacionados com os voos e aviões do problema *i*. $FCrcv_i$ é o rácio de custos dos voos e aviões do problema *i*. $\overline{FCrcv}_e$ é a média aritmética dos rácios de custos do voo e avião das soluções para todos os problemas incluídos numa execução *e* do sistema e *n* é o número total de execuções.

- **M17 - Rácio Médio de Recuperação de Custos da Tripulação:** Este rácio estabelece uma relação entre os custos relacionados com a tripulação inerentes à solução automática devolvida pelo sistema para um problema *i* (CC_i) e os custos previstos antes da resolução desse mesmo problema (OC_i). Maiores valores são melhores. Valores perto do 1 indicam que as soluções automáticas devolvidas pelo sistema possuem menores custos que o problema original. Valores perto do 0 indicam que as soluções possuem custos próximos aos do problema original. Se o valor for negativo, os custos das soluções automáticas ultrapassam os

custos iniciais.

$$CCrcv_i = \frac{CC_i}{CO_i} \tag{6.23}$$

$$\overline{CCrcv} = 1 - (\frac{\sum_{e=1}^{n}(\overline{CCrcv}_e)}{n}) \tag{6.24}$$

CC_i é o cálculo de todos os custos relacionados com a tripulação do problema i. $CCrcv_i$ é o rácio de custos da tripulação para o problema i. $\overline{FCrcv}_e$ é a média aritmética dos rácios de custos da tripulação das soluções para todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M18 - Número de Negociações até Aceitação:** De cada vez que o operador humano rejeita a solução encontrada, o sistema deve renegociar o problema e devolver uma nova solução. Esta métrica representa o número médio de negociações que o sistema executa para cada problema. Quanto menor for o número de negociações necessárias, melhor.

$$\overline{NN} = \frac{\sum_{e=1}^{n}(\overline{NN}_e)}{n} \tag{6.25}$$

$\overline{NN}_e$ é a média aritmética de número de negociações necessárias para chegar a uma solução que seja aceite pelo operador humano, para todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M19 - Qualidade Média Atribuída pelo Operador Humano:** Quando o operador humano aceita uma solução devolvida pelo sistema tem de especificar um valor quantitativo que traduz a qualidade da solução encontrada do seu ponto de vista. O operador humano pode especificar um valor no intervalo [0,10], sendo que maiores valores correspondem a melhores soluções. Este valor é calculado da seguinte forma:

$$\overline{Q_{op}} = \frac{\sum_{e=1}^{n}(\overline{Q_{op}}_e)}{n} \tag{6.26}$$

$\overline{Q_{op}}_e$ é a média aritmética da qualidade atribuída pelo operador humano às soluções, devolvidas pelo sistema e aceites pelo operador, para todos os problemas incluídos numa execução e do sistema e n é o número total de execuções.

- **M20 - Satisfação média do Operador Humano:** Estima-se que a solução que mais satisfaz o operador humano é aquela que conjuga maiores valores de utilidade global com os maiores valores da qualidade atribuída pelo primeiro. A satisfação é calculada através da multiplicação da qualidade média atribuída pelo operador humano ($\overline{Q_{op}}$) com a utilidade média global ($\overline{U_{global}}$). Esta última é multiplicada por 10 para colocar as métricas numa mesma escala.

$$\overline{S_{op}} = \frac{\overline{Q_{op}}}{\overline{U_{global}} \times 10} \tag{6.27}$$

6.5 Aprendizagem na Negociação Automática

Neste sub-capítulo são apresentadas as experiências efectuadas à solução de aprendizagem desenvolvida para a negociação automática entre agentes.

Os resultados apresentados dizem respeito às oito abordagens apresentadas: ST, NB, NQmA, NQmA-DOM, NSmA, NQmB, NSmB e NH. Os valores desta última abordagem são apenas discutidos no sub-capítulo 6.6. Nas experiências realizadas, foram efectuadas 100 execuções (n = 100) em todas as abordagens, com excepção da abordagem NH, em que foram efectuadas apenas 25 execuções (n = 25), pelas razões já referidas.

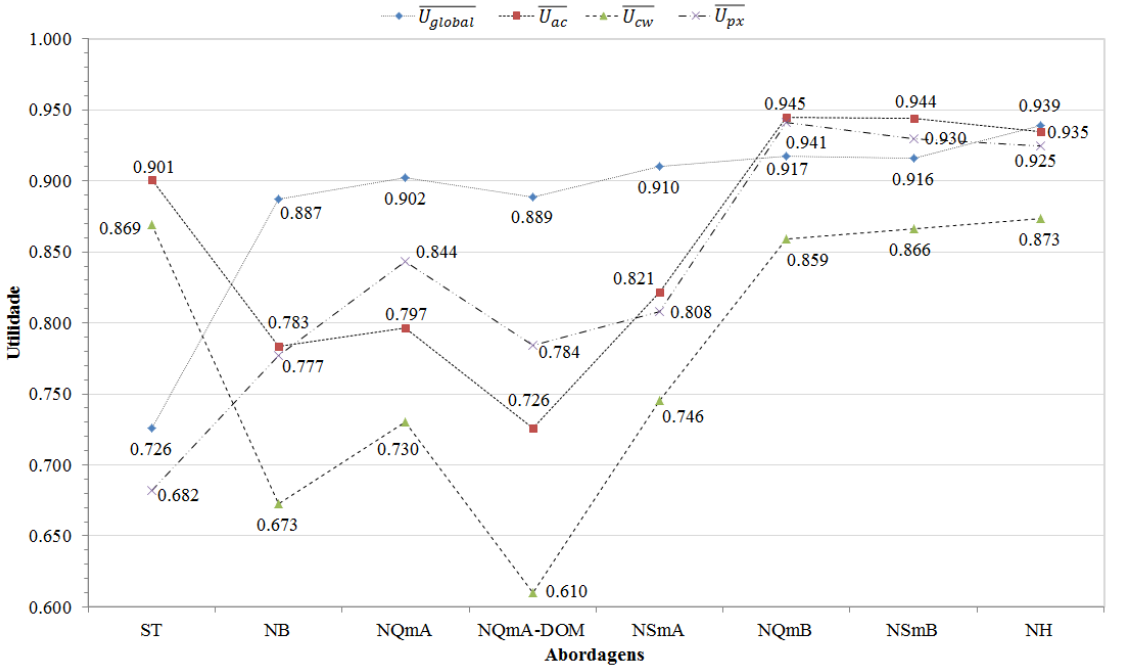

Figura 6.1: Utilidades Médias - Métricas M1, M2, M3 e M4

O gráfico da figura 6.1 apresenta a utilidade média global ($\overline{U_{global}}$) e as utilidades médias individuais dos agentes *Manager* ($\overline{U_{ac}}$, $\overline{U_{cw}}$ e $\overline{U_{px}}$) em cada abordagem.

De todas as abordagens referidas, é a abordagem ST aquela que possui pior utilidade global ($\overline{U_{global}} = 0.726$). Todavia, nessa mesma abordagem o agente *Aircraft Manager* possui uma utilidade individual elevada ($\overline{U_{ac}} = 0.901$), valor só ultrapassado nas abordagens NQmB e NSmB. Tal facto deve-se à resolução sequencial do problema, onde a perspectiva do avião é considerada em primeiro lugar e, por isso, sofre menos restrições.

Considerando o valor da diferença de utilidades, representado na figura 6.2, é também a abordagem ST aquela que possui a maior variação entre as utilidades dos agentes *Manager* ($\overline{Dif_U} = 0.219$), o que significa que é nesta abordagem que existe mais desigualdade, no que toca à utilidade individual das soluções, entre os agentes *Manager*.

Esta desigualdade é reduzida a partir da abordagem NB ($\overline{Dif_U} = 0.110$), onde se começa a utilizar a negociação automática entre agentes. A partir das experiências efectuadas pode deduzir-se

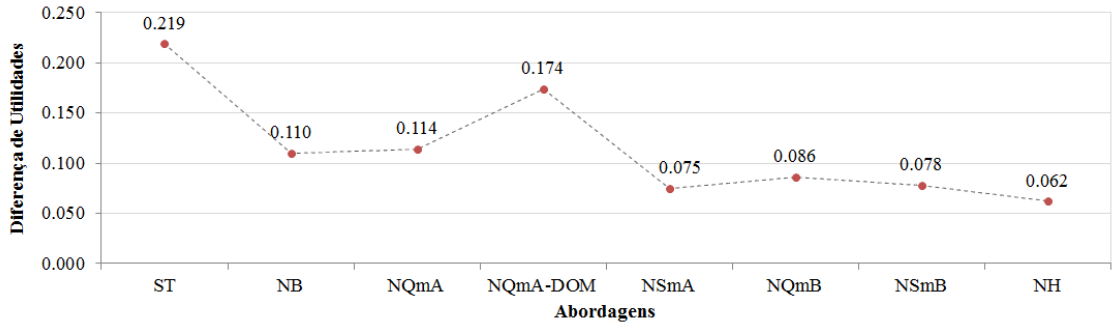

Figura 6.2: Diferença Média de Utilidades - Métrica M14

que, na abordagem NB, é o agente *Passenger Manager* aquele que mais lucra com a negociação, visto que é o único agente *Manager* a aumentar a sua utilidade individual. Considera-se este facto natural, uma vez que a perspectiva deste agente é a última a ser considerada na abordagem sequencial ST. Contudo, as utilidades global e individuais na abordagem NB são ainda relativamente baixas.

A abordagem NQmA-DOM foi a primeira solução a ser desenvolvida no âmbito desta dissertação. Os resultados obtidos não são animadores quando comparados com os resultados da abordagem NB. Não só a desigualdade entre as utilidades de cada agente *Manager* aumentou ($\overline{Dif_U} = 0.174$), como a própria utilidade global decresceu ($\overline{U_{global}} = 0.889$). Nesta abordagem, a acção do algoritmo de aprendizagem contém o elemento $action_{Domain}$. Na abordagem NQmA esse elemento não está incluído na representação da acção do algoritmo de aprendizagem, sendo esta a única diferença entre estas duas abordagens. Nos gráficos das figura 6.1 e 6.2 pode constatar-se a melhoria verificada na abordagem NQmA em relação a NQmA-DOM: todas as utilidades (global e individuais) sofreram um aumento e a diferença entre as utilidades individuais foi diminuída.

A abordagem NSmA demonstra melhores resultados que NQmA. Como descrito no subcapítulo 6.3, a diferença entres estas duas abordagens reside no algoritmo de aprendizagem por reforço utilizado. Usando o mecanismo *A* foi o algoritmo *Sarsa* que obteve os melhores resultados, embora tal não se verificasse com o mecanismo *B*, como iremos verificar.

Ambas as abordagens NSmA e NQmA apresentam uma melhoria face a NB, que não possui aprendizagem na negociação automática. A menor das utilidades individuais dos agentes *Manager* nestas abordagens foi $\overline{U_{cw}} = 0.730$, o que corresponde, por comparação com o valor de $\overline{U_{cw}} = 0.673$ da abordagem NB, a uma melhoria de 0.057 unidades. Contudo, alguns dos resultados obtidos nestas abordagens não são satisfatórios. A diferença entre utilidades individuais na abordagem NB ($\overline{Dif_U} = 0.110$) chega a ser mais satisfatória que na abordagem NQmA ($\overline{Dif_U} = 0.114$), embora seja uma diferença de apenas 0.004 unidades. No que toca à utilidade global, entre o melhor valor das abordagens que utilizam o mecanismo *A* ($\overline{U_{global}} = 0.910$) e o valor da utilidade global da abordagem NB ($\overline{U_{global}} = 0.887$), a variação corresponde a 0.023 unidades.

As experiências realizadas com o mecanismo *B* correspondem aos resultados das abordagens NQmB e NSmB. Em termos de utilidades, a melhoria conseguida com a utilização deste meca-

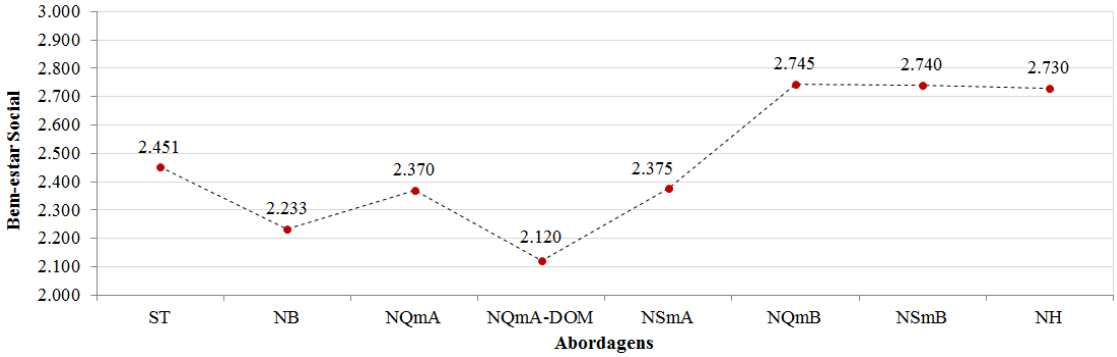

Figura 6.3: Bem-estar Social Médio - Métrica M5

nismo é significativa. A utilidade global obtida na abordagem NQmB foi a mais alta de qualquer uma das outras abordagens: $\overline{U_{global}} = 0.917$. Todas as utilidades individuais foram aumentadas. Na verdade, o valor mais baixo de utilidade individual obtido nesta abordagem ($\overline{U_{cw}} = 0.859$) é superior ao valor mais alto de utilidade individual obtido em qualquer uma das outras abordagens que utilize negociação automática entre agentes (NQmA com $\overline{U_{px}} = 0.844$ unidades). Quanto à diferença entre as utilidades individuais dos agentes *Manager*, a abordagem NQmB conseguiu o terceiro melhor valor ($\overline{Dif_U} = 0.086$) e a abordagem NSmB conseguiu o segundo melhor valor ($\overline{Dif_U} = 0.078$, apenas 0.003 unidades acima do melhor).

Entre as abordagens NQmB e NSmB existem diferenças mínimas no que toca às utilidades obtidas. A utilização do algoritmo *Sarsa* permitiu apenas melhorar o valor da utilidade individual do agente *Crew Member Manager* em 0.007 unidades. Esta utilidade corresponde à menor utilidade individual obtida em qualquer uma das abordagens. Como o cálculo de $\overline{Dif_U}$ utiliza os maiores e menores valores de utilidade individual, a abordagem NSmB conseguiu uma diferença entre utilidades individuais menor em relação à abordagem NQmB. Em todas as outras utilidades foi a abordagem NQmB que obteve resultados melhores.

No gráfico da figura 6.3 estão representados os valores do bem-estar social dos agentes *Manager*. Na análise a este valor deve também incluir-se a utilidade global do sistema para a mesma abordagem: a melhor situação será aquela em que ambos os valores $\overline{U_{global}}$ e $\overline{U_{bs}}$ são elevados. Isto significa que a solução é do agrado tanto da entidade central (utilidade global) como das entidades que constroem a solução (utilidades individuais). Deste ponto de vista, é fácil constatar que foram as abordagens NQmB e NSmB aquelas que obtiveram os melhores resultados, sendo que a abordagem NQmB supera o NSmB. As soluções encontradas conjugam uma elevada utilidade do ponto de vista global ($\overline{U_{global}} = 0.945$ na abordagem NQmB e $\overline{U_{global}} = 0.944$ na abordagem NSmB) com um elevado bem-estar social por parte dos agentes *Manager* ($\overline{U_{bs}} = 2.745$ na NQmB e $\overline{U_{bs}} = 2.740$ na NSmB). Melhor não foi conseguido por qualquer outra abordagem. O terceiro maior valor de bem-estar social corresponde à abordagem ST ($\overline{U_{bs}} = 2.451$). Contudo, o valor de utilidade global da abordagem ST ($\overline{U_{global}} = 0.726$) é o menor de todas as abordagens. Observe-se que mesmo a diferença entre utilidades individuais, na abordagem ST, é a mais elevada de todas, o que realça o desequilíbrio de utilidades individuais entre os agentes *Manager*.

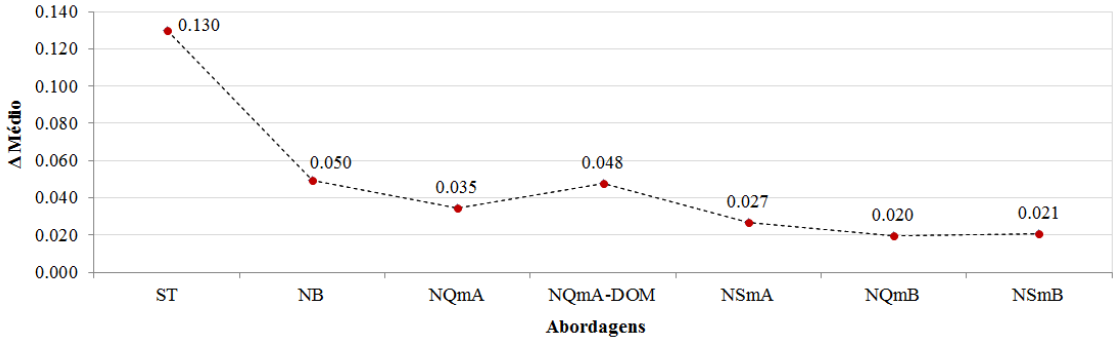

Figura 6.4: Δ Médio para a Solução Óptima - Métrica M6

O $\Delta(optimal)$ de cada abordagem pode ser consultado no gráfico da figura 6.4. Mais uma vez, são as abordagens NQmB e NSmB aquelas que possuem os melhores resultados. Na abordagem NQmB, o $\Delta(optimal)$ foi reduzido para 0.020 unidades. Na abordagem com negociação automática sem aprendizagem NB esse valor correspondia a 0.050 unidades. Foi na abordagem CST que se obteve a maior diferença, que correspondia a 0.130 unidades.

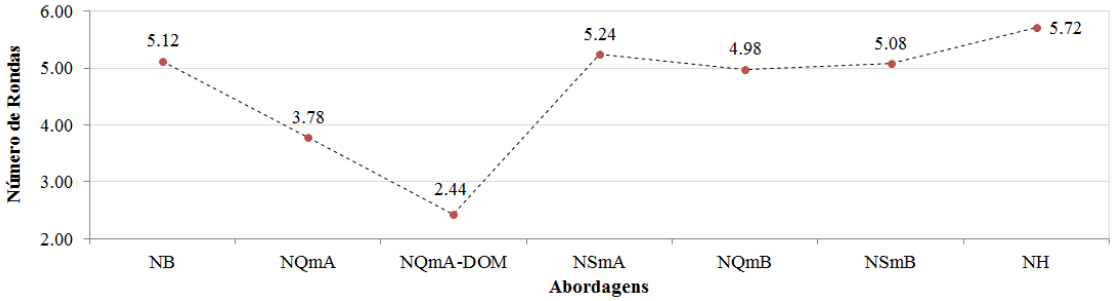

Figura 6.5: Número Médio de Rondas - Métrica M13

No gráfico da figura 6.5 pode ser consultado o número de rondas por negociação necessárias para encontrar a solução final. Considerando apenas as abordagens com um valor de $\overline{NN}$ superior à abordagem NB, a diferença de valores entre a abordagem NB e cada um destas é mínima, pelo que se considera que nenhum mecanismo prejudicou a negociação em termos de tempo necessário para encontrar uma solução.

No gráfico da figura 6.6 é possível consultar os rácios de recuperação de custos e atrasos. Os melhores valores encontram-se nas abordagens NQmB, NSmB e NSmA. São, no entanto, as abordagens NQmB e NSmB as que conseguiram, de uma forma global, um melhor desempenho a nível de rácios de recuperação. No rácio de recuperação de custos de voo e avião, a abordagem NSmA ($\overline{FCrcv} = 0.177$) acaba por distanciar-se negativamente às abordagens NQmB ($\overline{FCrcv} = 0.286$) e NSmB = ($\overline{FCrcv} = 0.328$).

Embora o rácio de recuperação seja mais alto para os atrasos da perspectiva do avião, foi nos custos das perspectivas do avião e tripulação que se sentiu mais diferença entre os valores das diferentes abordagens. No rácio de recuperação de custos da perspectiva do avião, na abordagem ST os valores são negativos, o que significa que as soluções automáticas representavam mais custos

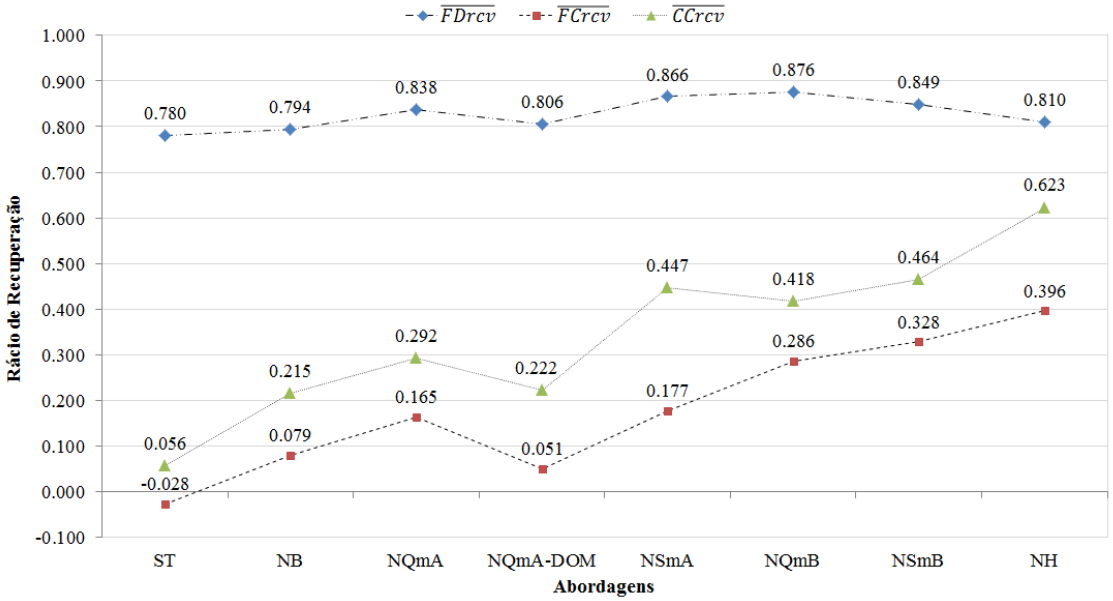

Figura 6.6: Rácios Médios de Recuperação de Atrasos e Custos - Métricas M15, M16 e M17

para a companhia aérea, na perspectiva do avião, do que não resolver o problema. Nas abordagens NQmB e NSmB, esse rácio foi elevado em cerca de 30%. Já quanto aos custos de tripulação, as abordagens NQmB e NSmB elevaram o rácio de recuperação em cerca de 40%, face à abordagem ST.

Na indústria do transporte aéreo utiliza-se uma métrica que mede os atrasos de voos, denominada *Pontualidade*. Para o cálculo desta métrica são apenas considerados voos com atrasos superiores a 15 minutos. No gráfico da figura 6.7 pode encontrar-se a percentagem de voos com atraso superior a quinze minutos. Mais uma vez, são as abordagens NQmB e NSmB que possuem resultados superiores. Na abordagem NSmB conseguiu-se que nenhum voo sofresse um atraso superior a 15 minutos, em média, o que também se traduz numa utilidade individual alta para a perspectiva do avião, como é possível analisar no gráfico da figura 6.1.

Os custos médios para os passageiros associados às novas soluções estão representados na figura 6.8. Pode verificar-se que foi a abordagem NQmA aquela que mais reduziu, de entre todas, os custos dos passageiros, seguido da abordagem NQmB.

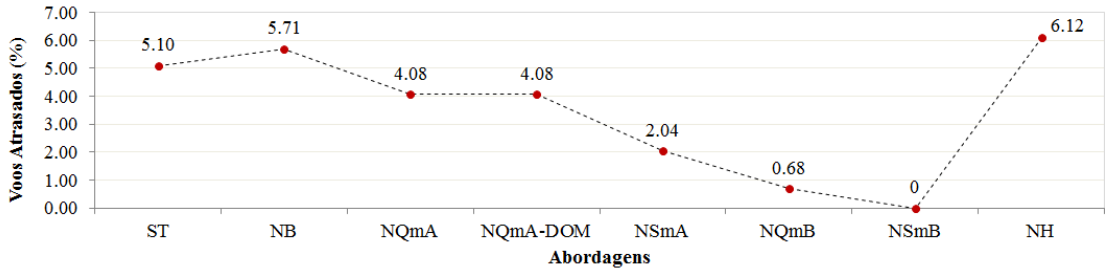

Figura 6.7: Atraso Médio Superior a 15 minutos dos Voos - Percentagem de voos com atrasos médios (métrica M7) superiores a 15 minutos.

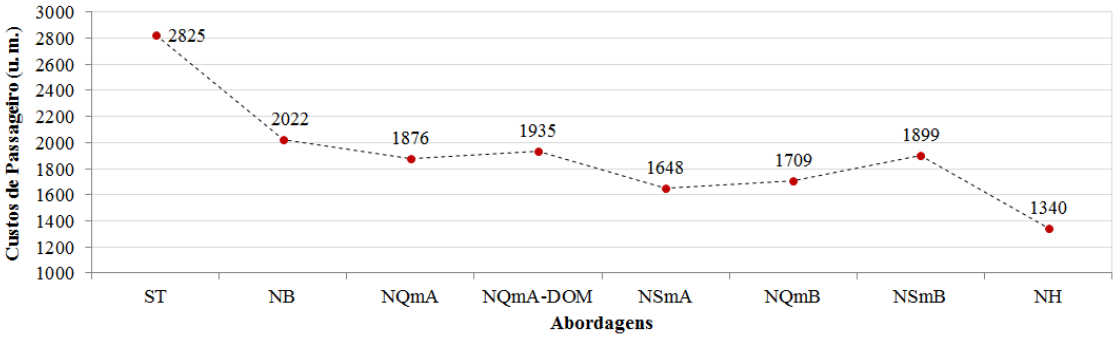

Figura 6.8: Custos Médios dos Passageiros - Métrica M12

Após a avaliar os gráficos demonstrados, pode concluir-se que foram as abordagens NQmB e NSmB aqueles que conseguiram um melhor desempenho global.

6.5.1 Interpretação dos Resultados

O primeiro mecanismo de aprendizagem desenvolvido é utilizado na abordagem NQmA-DOM. A arquitectura deste mecanismo está representada na figura 4.1. Como foi descrito no capítulo 4, a representação da acção no algoritmo de aprendizagem desta abordagem contém um elemento, denominado $action_{Domain}$, que filtra as soluções parciais de acordo com o plano de domínio. No sub-capítulo 4.2.1 foi referido que, em experiências preliminares, era usual a ocorrência de situações em que o agente *Manager* que ganhava a negociação na primeira ronda tendia a ser o vencedor de todas as rondas e, consequentemente, de toda a negociação.

A abordagem NQmA-DOM foi o que registou o menor número de rondas necessárias para encontrar a solução final. Contudo, as métricas apresentadas permitiram constatar que, das abordagens que foram desenvolvidas neste dissertação, é no NQmA-DOM que se obtém os piores resultados. Destas informações confirma-se uma das hipóteses delineadas no final do sub-capítulo 4.2.1: o elemento $action_{Domain}$ restringia de tal forma as opções do agente *Manager* que este ficava sem soluções que pudesse apresentar ao agente *Supervisor*. O agente *Manager* que ganhasse a primeira ronda tendia a ser o vencedor da negociação, pois não encontrava oposição. Este comportamento tinha um efeito muito negativo na negociação, pois os agentes não procuravam soluções que mais agradassem o agente *Supervisor*. Os agentes que propunham a solução vencedora na primeira ronda tendiam a vencer as rondas seguintes não porque a sua proposta fosse demasiado boa, mas sim porque os outros agentes *Manager* não eram capazes de propor nenhuma solução. Daí os maus resultados das soluções nas métricas apresentadas anteriormente.

A remoção deste elemento, que corresponde ao plano, de domínio da acção do algoritmo constitui a alteração que distingue a abordagem NQmA da NQmA-DOM. A melhoria é evidente, já que a abordagem NQmA obteve melhores resultados que a NQmA-DOM em todas as métricas. A excepção está apenas no número de rondas de negociação necessárias para encontrar a solução, o que, como foi anteriormente explicado, não é positivo.

Os resultados da abordagem NSmA superam os da abordagem NQmA. A única diferença entre ambas é o algoritmo utilizado: no primeiro é utilizado o algoritmo *Sarsa* enquanto que no segundo utiliza-se o algoritmo *Q-Learning*. Contudo, a partir das experiências realizadas não é possível nem sustentar o facto de o algoritmo *Sarsa* ter apresentado melhores resultados, nem a afirmação de que o algoritmo *Sarsa* é mais indicado para o contexto do sistema MASDIMA. A diferença entre as abordagens NQmB e NSmB reside também no algoritmo utilizado, e, todavia, considera-se que foi o algoritmo *Q-Learning* o algoritmo com melhor desempenho, embora a diferença não fosse muito acentuada. Todavia, o objectivo desta dissertação não se prendia com a comparação entre o desempenho de algoritmos de aprendizagem. Consideram-se ambos os algoritmos bastante adequados para o contexto de aprendizagem existente no MASDIMA.

A análise que se havia feito previamente ao mecanismo, aquando do operador $action_{Domain}$, permitiu pôr em causa a função dos motores de aprendizagem entre os agentes *Manager* que fazem parte da arquitectura do mecanismo *A*. Outra das hipóteses delineadas no final do sub-capítulo 4.2.1 apontava para a existência de um efeito muito restritivo na actividade entre os agentes *Manager*. Por um lado, os agentes tinham de respeitar as restrições que outro agente *Manager* impunha quando lhes pedia a sua colaboração. Por outro lado, existia a acção do algoritmo de aprendizagem que restringia ainda mais o já pequeno conjunto de soluções possíveis. Esta conjunção de restrições impedia os agentes de completar a solução de quem a pedia, pois ficam sem soluções que respeitassem essa conjunção. O primeiro agente via-se impossibilitado de apresentar uma solução integrada ao agente *Supervisor* e, mais uma vez, aquele que ganhasse a primeira ronda tendia a ser o vencedor da negociação, pois raramente encontrava oposição. Ainda que os resultados destas abordagens sejam mais satisfatórios do que na abordagem NB, os valores da utilidade global e individual de cada agente ou ainda o valor do bem-estar sugerem a inexistência de uma procura eficiente de soluções, por parte dos agentes *Manager*.

A alteração da arquitectura do mecanismo *A* deu origem ao mecanismo *B*, representado na figura 4.3, que é utilizado nas abordagens NQmB e NSmB. Dos resultados obtidos nestas abordagens conclui-se que este é o mecanismo mais adaptado ao ambiente simultaneamente cooperativo e competitivo onde os agentes *Manager* se inserem. Como foi descrito no sub-capítulo 4.2.2, neste mecanismo os agentes que pedem a cooperação de outros têm mais consciência da capacidade dos seus colaboradores e têm a possibilidade de agir em conformidade com essas capacidades. Os resultados demonstram que as soluções vencedoras têm uma elevada utilidade não apenas do ponto de vista global nem unicamente para o agente que a propôs. Todos os agentes possuem valores elevados de utilidade nas soluções, o que significa que todos os agentes ganham com a solução, mesmo não sendo os vencedores da negociação.

Embora exista uma competição, a cooperação é tal que nenhum dos agentes perde em cooperar com os outros. Não só o bem-estar social dos agentes *Manager* é elevado, como a própria qualidade das soluções, para o contexto real, melhorou significativamente. Em teoria de jogos existe o termo *win-win* para denominar jogos cujos participantes lucram sempre de uma forma ou de outra. Poderíamos, assim, apelidar o mecanismo *B* como um mecanismo *win-win*, adequado para ambientes simultaneamente cooperativos e competitivos.

6.6 *Human-in-the-Loop*

Neste sub-capítulo são apresentadas as experiências efectuadas à solução que permite a interacção do operador humano com o sistema. São aqui discutidas apenas as abordagens NQmB e NH, visto que o objectivo é apresentar resultados que permitam caracterizar a funcionalidade *Human-in-the-Loop*. Assim, optou-se por comparar o desempenho entre as abordagens cuja única diferença seja a existência dessa funcionalidade.

Nas figuras 6.1 e 6.2 pode observar-se que tanto o valor da utilidade global como o valor da diferença de utilidades da abordagem NH ($\overline{U_{global}} = 0.939$ e $\overline{Dif_U} = 0.061$) são melhores do que os valores correspondentes da abordagem NQmB ($\overline{U_{global}} = 0.917$ e $\overline{Dif_U} = 0.086$). Contudo, o valor do bem-estar social dos agentes acaba por ser menor na abordagem NH ($\overline{U_{bs}} = 2.730$) do que na abordagem NQmB ($\overline{U_{bs}} = 2.745$).

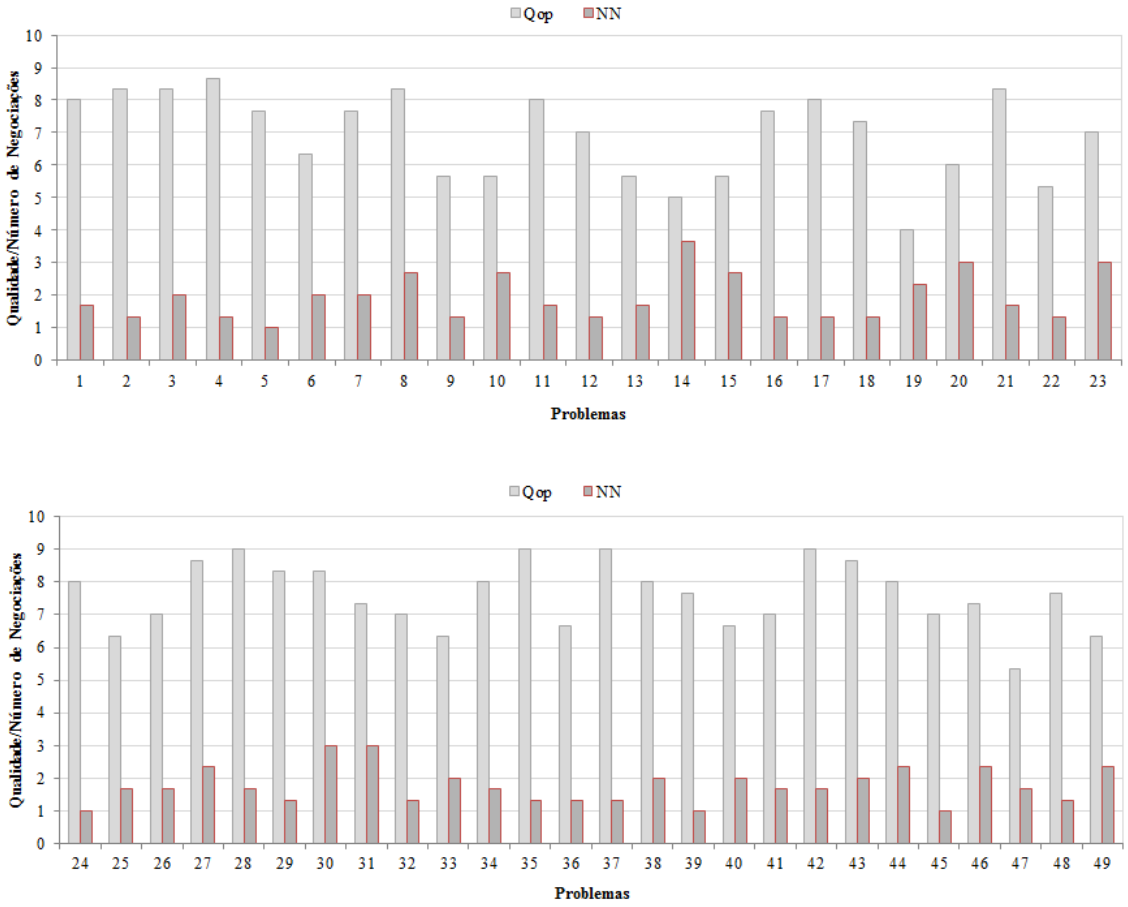

Figura 6.9: Relação $\overline{Q_{op}}$ / $\overline{NN}$ - Relação entre as métricas M19 e M18 na abordagem NH.

A partir dos gráficos das figuras 6.6 pode verificar-se que, para as perspectivas do avião e tripulação, houve uma recuperação de custos mais acentuada na abordagem NH do que em qualquer outra ($FCrcv = 0.396$ e ($CCrcv = 0.623$)). No gráfico da figura 6.8 pode também verificar-se que foi esta a abordagem que devolveu a melhor resposta para os custos da perspectiva dos passageiros ($\overline{PC} = 1340$ u.m.). Já no rácio de recuperação de atrasos da perspectiva do avião houve uma

pequena diminuição ($FDrcv = 0.810$). Essa diminuição vai de encontro à percentagem de voos atrasados mais do que 15 minutos, que pode ser encontrada no gráfico da figura 6.7. A abordagem NH registou a mais alta percentagem, com cerca de 6.12% de voos atrasados mais do que 15 minutos.

No gráfico da figura 6.9 apresentam-se os valores médios de $\overline{Q_{op}}$ e $\overline{NN}$ para cada problema, utilizando a abordagem NH. O ideal seria conjugar o maior valor de $\overline{Q_{op}}$ com o menor valor de $\overline{NN}$. O valor médio de negociações necessárias para chegar a uma solução aceite pelo operador humano, para todos os problemas considerados, é $\overline{NN} = 1.843$, o que significa que foi necessário renegociar cada problema, em média, 1.843 vezes. Nas experiências efectuadas não houve nenhum problema a ser renegociado mais do que 7 vezes e cerca de 50% dos problemas foram aceites logo na primeira negociação. A qualidade média atribuída pelo operador (equação 6.26) às soluções aceites foi de $\overline{Q_{op}} = 7.293$.

Note-se que são os problemas com menor número de negociações (como o problema 5 com $\overline{NN} = 1.000$) que obtêm os melhores valores de qualidade ($\overline{Q_{op}} = 7.666$). Nos problemas onde foi necessário um maior número de negociações (como no problema 14 com $\overline{NN} = 3.666$), a qualidade atribuída pelo operador humano tende a ser menor ($\overline{Q_{op}} = 5.000$).

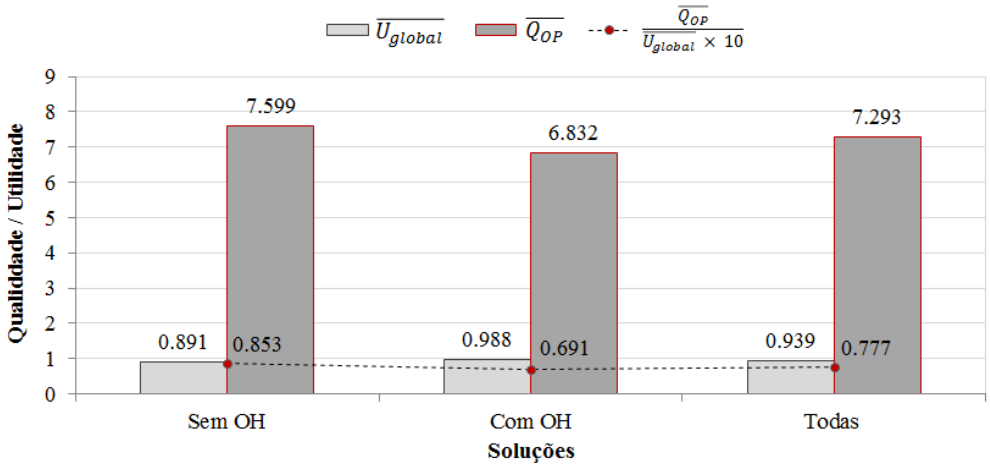

Figura 6.10: Relação $\overline{Q_{op}}$ / $\overline{U_{global}}$ - Relação entre as métricas M19 e M1.

No gráfico da figura 6.10 pode observar-se uma comparação entre soluções aceites que foram obtidas sem a intervenção do operador humano (classe *sem OH*) e entre soluções obtidas com a intervenção do operador humano (classe *com OH*). Nos casos sem intervenção do operador humano foi efectuada apenas uma negociação com recurso aos valores por omissão dos parâmetros α e β (tabela B.1). Este gráfico apresenta a relação entre a qualidade média atribuída pelo operador humano e utilidade média global (multiplicada por 10). A intervenção do operador humano implica a alteração dos parâmetros α e β, através da estratégia descrita no capítulo 5, e a negociação automática do problema mais do que uma vez. Os resultados obtidos demonstram que a utilidade média global é maior em soluções avaliadas através de valores alterados dos parâmetros α e β ($\overline{U}_{global} = 0.988$), o que está de acordo com os valores de utilidade global do gráfico da figura

6.1. Contudo, a qualidade média atribuída pelo operador é maior nas soluções em que não houve intervenção do operador humano $\overline{Q}_{op} = 7.599$.

6.6.1 Interpretação de Resultados

O aumento do valor de utilidade global na abordagem NH pode não representar um aumento real da utilidade global das soluções. Na abordagem NH é feita a variação dos parâmetros α e β da função de avaliação. Logo, a função de utilidade global utilizada na abordagem NH pode não corresponder à função de utilidade global utilizada nas restantes abordagens. Esta alteração leva a que uma mesma solução possa corresponder a diferentes valores de utilidade quando consideramos a abordagem NH e outras abordagens. Assim, não é possível retirar uma conclusão relativamente à qualidade global das soluções. Por essa razão, não se apresentou informação quanto à métrica $\Delta(optimal)$ para esta abordagem.

Não se pode deixar de reparar que, após a alteração dos parâmetros α e β, as utilidades médias das soluções tende a subir, ao invés de descer. Através das experiências realizadas não foi possível apontar uma razão para este facto.

A utilidade individual de cada agente *Manager* é calculada da mesma forma em todas as abordagens. Podemos observar, no gráfico da figura 6.1, que a utilidade dos diferentes agentes não foi comprometida. Embora as utilidades dos agentes *Passenger Manager* e *Aircraft Manager* tenham decrescido em 0.016 e 0.010 unidades respectivamente, a utilidade do agente *Crew Member Manager* aumentou em 0.014 unidades. Tendo a diferença de utilidades sido reduzida, é possível afirmar que a interacção do operador humano com o sistema não se traduziu num desequilíbrio de importância atribuída a cada perspectiva, muito embora os parâmetros α sofressem alterações durante as negociações. É verdade que o bem-estar social (ver figura 6.3), na abordagem NH, decresceu 0.015 unidades em relação à abordagem NQmB. Contudo, esse valor continua bastante acima da média.

Através destes valores verifica-se que a interacção do operador humano não teve um impacto negativo na qualidade das soluções para os diferentes agentes. Foi possível até reduzir as diferenças entre a qualidade das soluções para os agentes *Manager*. O pequeno aumento que o número médio de rondas necessárias para encontrar a solução numa negociação sofreu, na abordagem NH ($\overline{NR} = 5.72$), sugere que a alteração dos parâmetros da função de avaliação não tem um impacto directo na aprendizagem dos agentes *Manager*, que conseguem continuar a devolver soluções com bastante utilidade global e individual. Embora fosse necessário submeter os mecanismos dos agentes *Manager* a mais experiências para obter resultados mais conclusivos, através das experiências realizadas não foi possível identificar interferências entre os mecanismos do agente *Supervisor* e dos agentes *Manager*.

Ainda sobre a abordagem NH, é interessante referir o seguinte. Os resultados obtidos a nível de custos e atrasos para as diferentes perspectivas, nas soluções avaliadas com a intervenção do operador humano, demonstram a atitude de um operador humano em relação aos problemas. A função do sistema é minimizar todos os custos e atrasos da solução. Já o operador humano tende a avaliar as soluções não tanto pela utilidade global que esta representava, mas sim pelos planos para

cada perspectiva que a solução ditava. Todas as soluções que envolvessem planos para as perspectivas como o cancelamento do voo ou a contratação de outra companhia para a realização de um voo para os passageiros era avaliada com uma qualidade tendencialmente mais baixa, mesmo que os valores de utilidade fossem superiores a uma outra solução que não indicasse esses planos.

Outro indício notado está relacionado com os atrasos dos voos. Como foi anteriormente referido, na indústria do transporte aéreo costuma-se considerar um avião atrasado apenas se o atraso for superior a 15 minutos (métrica *Pontualidade*). Após a intervenção do operador humano, as soluções vencedoras possuem maiores atrasos na perspectiva do avião. A percentagem de voos com atrasos superiores a 15 minutos é a maior de todas as abordagens. Este aumento reflecte-se no rácio de recuperação dos atrasos da perspectiva do avião, que diminuiu, e ainda na utilidade individual do agente *Manager*.

O gráfico da figura 6.10 corrobora a análise anterior, pois a satisfação do operador humano em relação às soluções devolvidas após a sua intervenção é menor do que a satisfação em relação às soluções devolvidas sem a sua intervenção, apesar de as primeiras possuírem melhor utilidade global. De facto, verifica-se que, quando existiu um maior número de interacções com o operador humano (mais negociações) a qualidade atribuída por este foi pior. Existem algumas hipóteses para explicar este facto. Por um lado, uma maior alteração dos parâmetros da função de utilidade pode estar a classificar as soluções de uma forma errada, em comparação com os objectivos do operador humano. Por outro, a qualidade atribuída pode ter uma relação inversa com o tempo decorrido até à devolução de uma proposta que finalmente se adapte ao contexto e também com a atenção que o operador humano tem que despender e que cresce consoante o número de interacções necessárias.

Constatou-se ainda que cerca de 50% das soluções foi aceite sem necessidade de intervenção do operador humano, tendo as mesmas sido mais bem classificadas do que as restantes. A partir dos dados apresentados é possível concluir que, em média, o sistema MASDIMA devolve soluções adequadas ao problema mesmo sem a intervenção do operador humano. Pode, no entanto, existir um desfasamento de objectivos entre a função de utilidade global do sistema (com valores por omissão dos parâmetros α e β) e os objectivos do operador humano que participou nesta experiência. O número médio de renegociações é, sensivelmente, de duas por problema. Significa isto que, geralmente, o operador humano necessita de efectuar apenas uma avaliação (que foi avaliada pelo sistema com recurso aos valores por omissão dos parâmetros β e α), para que o sistema seja capaz de corresponder às suas preferências. Conclui-se, deste facto, que a estratégia de adaptação dos parâmetros α e β permite obter uma boa resposta por parte do sistema à avaliação do operador humano.

6.7 Resumo

Neste capítulo foram demonstrados os resultados das experiências efectuada no âmbito desta dissertação. Foram analisadas 20 métricas em 7 abordagens diferentes. Cada abordagem corresponde a um método de resolução de problemas distinto. As abordagens NQmA-DOM, NQmA,

NSmA, NQmB e NSmB permitem testar as funcionalidades desenvolvidas para a negociação automática entre agentes. A abordagem NH permite testar a funcionalidade *Human-in-the-Loop*. Relativamente à aprendizagem na negociação automática, as conclusões mais importantes a retirar deste capítulo são:

- O mecanismo de aprendizagem *B*, utilizado nas abordagens NQmB e NSmB, permite obter os melhores resultados em todas as métricas utilizadas, tendo-se destacado como o mecanismo, desenvolvido nesta dissertação, mais adaptado ao ambiente simultaneamente cooperativo e competitivo dos agentes *Manager*.

- Considera-se que a utilização do mecanismo *B* na negociação automática traz vantagens a todos os agentes. Não está em causa apenas uma vantagem individual, já que se conseguiu conjugar valores máximos de utilidades individuais com valores máximos da utilidade global do sistema.

- O mecanismo de aprendizagem *A*, utilizado nas abordagens NQmA-DOM, NQmA e NSmA, não permite uma boa cooperação entre os agentes *Manager* e impossibilita, muitas vezes, os agentes *Manager* de competir na negociação automática.

Quanto à interacção do operador humano com o sistema, as experiências efectuadas permitem concluir o seguinte:

- O sistema MASDIMA consegue reagir bem à avaliação do operador humano, pois, geralmente, os problemas não são negociados mais do que duas vezes, pelo que a estratégia de adaptação do sistema está adequada ao funcionamento deste.

- O sistema MASDIMA consegue devolver soluções sem a intervenção do operador humano que são do agrado deste, o que significa que o sistema consiste numa boa ferramenta automática para a gestão de rupturas.

- A intervenção do operador humano na função de utilidade nem sempre resulta numa solução com mais qualidade para este, apesar dos valores elevados de utilidade global da solução, o que sugere um desfasamento entre os objectivo do sistema e do operador humano.

Capítulo 7

Conclusão

No presente capítulo são apresentadas as conclusões em relação ao trabalho desenvolvido e descrito neste documento. São ainda sugeridos alguns tópicos de investigação e desenvolvimento que podem ser explorados no seguimento desta dissertação.

7.1 Satisfação dos Objetivos

O objectivo do trabalho incluía dois tópicos distintos. Por um lado, existia a necessidade de desenvolver um processo de aprendizagem adaptado ao contexto competitivo e cooperativo dos agentes *Manager* que lhes permitisse aprender, durante as rondas das negociações automáticas, as preferências do agente *Supervisor*. Por outro, o grau de automatização do sistema MASDIMA exigia a possibilidade de validar e avaliar as soluções finais por um operador humano, de forma a reduzir a relutância social face ao sistema.

Na área de aprendizagem na negociação automática foram desenvolvidos dois mecanismos[1] de aprendizagem distintos: *A* e *B*. O mecanismo *A* foi o primeiro mecanismo a ser desenvolvido. A sua primeira versão não obteve resultados satisfatórios. Embora tivessem sido efectuadas alterações de forma a melhorar os resultados obtidos, este mecanismo continuava a apresentar indícios de não ser perfeitamente adaptado ao ambiente dos agentes *Manager* que, como já foi referido, é simultaneamente cooperativo e competitivo. O mecanismo *B* surgiu no seguimento do mecanismo *A*. A filosofia deste mecanismo distingue-se do mecanismo *A* por exigir mais consciência do agente que o implementa relativamente aos agentes que consigo colaboram. Os resultados obtidos com este mecanismo de aprendizagem, no contexto do sistema MASDIMA, permitem afirmar que se conseguiu o desenvolvimento de um mecanismo adaptado a este ambiente cooperativo e competitivo. As soluções obtidas pelo sistema com a implementação do mecanismo *B* aumentaram a sua qualidade em relação às soluções de versões anteriores do sistema, ao mesmo tempo

[1] Nome atribuído, no âmbito desta dissertação, ao processo que permite aos diferentes agentes *Manager* aprender a melhorar a avaliação atribuída pelo agente *Supervisor* às suas propostas.

que aumentaram as utilidades individuais dos agentes que compõe a solução. Foi, por isso, considerado um mecanismo *win-win*. Este tópico corresponde a uma das principais contribuições desta dissertação.

Quanto ao tópico *Human-in-the-Loop*, a funcionalidade desenvolvida permite a interacção do operador humano com o sistema sem o tornar dependente dela, ou seja, o sistema mantém a sua autonomia na negociação automática entre agentes. A solução vencedora da negociação tem de ser validada e classificada pelo operador humano. Se este rejeitar a solução, o sistema tem de renegociar o problema e encontrar uma solução adaptada à classificação atribuída pelo operador humano à solução rejeitada. Esta funcionalidade permitiu validar as soluções devolvidas pelo sistema. O operador humano que participou nas experiências efectuadas validou cerca de 50% das soluções após a primeira negociação do problema, o que significa que, nestes casos, não foi necessária a renegociação do problema. As soluções obtidas sem a intervenção do operador humano conseguiram uma boa avaliação (cerca de 7.6 numa escala de 0 a 10) por parte deste.

O sistema reage bem à avaliação atribuída pelo operador humano às soluções quando estas são rejeitadas. Em média, foi necessária apenas uma rejeição do operador humano por problema (duas negociações) para que o sistema devolvesse uma solução que o operador humano aceitasse. Esta funcionalidade constitui outra das principais contribuições científicas desta dissertação.

Os objectivos desta dissertação foram cumpridos, tendo-se obtido resultados interessantes que contribuem para o desenvolvimento da investigação académica, quer na área de sistemas multi-agente, quer na área de interacção de humanos com sistemas automáticos. O trabalho desenvolvido contribui ainda para a inovação da indústria, nomeadamente para a indústria do transporte aéreo. Não obstante, existem ainda alguns tópicos que seria interessante explorar e que são apresentados de seguida.

7.2 Trabalho Futuro

Durante o desenvolvimento desta dissertação surgiram algumas hipóteses e questões que se consideram relevantes investigar e desenvolver em trabalho futuro.

Em relação ao mecanismo de aprendizagem desenvolvido para a negociação automática entre agentes, crê-se que os conceitos do mecanismo *B* (secção 4.2.2) podem ser aplicados noutros ambientes simultaneamente cooperativos e competitivos fora do domínio do sistema MASDIMA. Contudo, para confirmar esta hipótese, seria necessário generalizar o mecanismo e adaptá-lo a outros contextos. Tal não foi realizado no âmbito desta dissertação.

Relativamente à funcionalidade *Human-in-the-Loop*, dos resultados obtidos nas experiências realizadas, foi identificado um desfasamento, em certas ocasiões, entre o objectivo do operador humano e o objectivo do sistema. Seria assim pertinente elaborar um estudo que permita verificar se os parâmetros incluídos na função de utilidade usada no sistema MASDIMA se encontram desajustados dos parâmetros avaliados num contexto real pelo operador humano. Nomeadamente, e uma vez que se verificou que nem sempre eram os custos e atrasos das várias perspectivas

a maior preocupação do operador humano, seria necessário estudar eventuais componentes de natureza mais social que não estejam englobados na actual função de utilidade.

A adaptação dos valores preferidos do agente *Supervisor* não foi explorada. A alteração destes valores implica um estudo aprofundado sobre o impacto de tal alteração no desempenho dos agentes *Manager*, pois os seus mecanismos de aprendizagem dependem desta informação. Esta constitui a última das linhas a explorar identificadas no desenvolvimento desta dissertação.

Anexo A

Mecanismo de Aprendizagem

Tabela A.1: Causas de Problemas - Identificadas a partir de entrevistas aos membros do controlo operacional aéreo da TAP Portugal. Correspondem aos valores possíveis do elemento *cause* de um estado em qualquer um dos mecanismos de aprendizagem apresentados.

Valor	Descrição
airp	Instalações e serviços fornecidos pelo Aeroporto. A título de exemplo, estacionamento indisponível, questões relacionadas com imigração.
atc	*En-route*, restrições de Controlo de Tráfego Aéreo e condições meteorológicas no aeroporto de destino.
comm	Atrasos ou falhas de presença relacionadas com passageiros.
crot	Atrasos ou falhas de presença relacionadas com tripulantes.
hand	Eventos relacionados com a manutenção de voo no que toca ao embarque de passageiros ou carregamento de bagagem.
induty	Problemas como doenças, acidentes relacionados com os tripulantes durante o voo ou durante a estadia.
maint	Avarias no avião.
meteo	Condições meteorológicas no aeroporto de origem.
rot	Problemas relacionados com rotação de aviões entre os voos, bem como atrasos à chegada.
rules	Violação de leis laborais, tais como exceder o tempo máximo de trabalho diário.
sec	Identificação de bagagem, procura e recolha da mesma depois do voo.
sign	Tripulante não se apresenta para o serviço.
other	Qualquer outro evento não incluído nos anteriores.

Tabela A.2: Planos de Domínio da Perspectiva *Passenger* - Correspondem aos valores para o elemento $action_{Domain}$ de uma acção do mecanismo de aprendizagem A na perspectiva *Passenger*.

Valor	Descrição
change_flt_change_airl	Colocar o passageiro num outro voo de outra companhia aérea.
change_flt_same_airl	Colocar o passageiro num outro voo da mesma companhia.
keep_same_flt	Manter o passageiro no voo.

Tabela A.3: Planos de Domínio da Perspectiva *Aircraft* - Correspondem aos valores para o elemento $action_{Domain}$ de uma acção do mecanismo de aprendizagem *A* na perspectiva *Aircraft*.

Valor	Descrição
cancel	Cancelamento do voo.
delay	Atraso do voo num específico número de minutos.
exchange	Alteração do avião por outro avião de outro voo.
other	Alugar tripulação e avião a outra companhia.

Tabela A.4: Planos de Domínio da Perspectiva *Crew Member* - Correspondem aos valores para o elemento $action_{Domain}$ de uma acção do mecanismo de aprendizagem *A* na perspectiva *Crew Member*.

Valor	Descrição
accept_delayed_crew	Utilizar a tripulação atrasada no voo, o que implica aceitar o atraso no voo.
exchange_crew	Substituir a tripulação atrasada por outra tripulação de outro voo.
other	Alteração do avião ou cancelamento do voo.
proceed_without_crew	Prosseguimento do voo sem os membros de tripulação que faltam.
use_crew_on_vacation	Utilização de um tripulante que está de férias.
use_dayoff_crew	Utilização de um tripulante que está de folga.
use_free_time_crew	Utilização de um tripulante que está com tempo disponível.
use_reserve_crew	Utilização de um tripulante que está em reserva.

Tabela A.5: Valores dos Elementos de uma Acção - Valores dos elementos de uma acção, em qualquer um dos mecanismos de aprendizagem apresentados, que dizem respeito aos atributos de custos e atrasos das propostas de solução.

Valor	Descrição
Inc	Aumentar - o valor do parâmetro deverá ser maior que o valor desse mesmo parâmetro na última proposta.
Keep	Manter - o valor do parâmetro deverá ser o mesmo da última proposta.
Dec	Diminuir - o valor do parâmetro deverá ser menor que o valor desse mesmo parâmetro na última proposta.

Tabela A.6: Pontuação da Classificação (mecanismo *B*) - Pontuação atribuída a cada valor de classificação para sumariar a situação global dos agentes *Manager*.

Classificação	Pontuação
Very Bad	2.0
High	1.0
Ok	0.0
Low	0.5

Anexo B

Human-in-the-Loop

Tabela B.1: Parâmetros de Avaliação Global de Soluções - Valores por omissão para os parâmetros de avaliação de soluções do agente *Supervisor*.

Parâmetro	**Valor**
α_{ac}	0.33
α_{cw}	0.33
α_{px}	0.33
β_{cost_ac}	0.33
β_{cost_cw}	0.33
β_{cost_px}	0.33
β_{delay_ac}	1.00
β_{delay_cw}	0.33
β_{delay_px}	0.66
max_{cost_ac}	1.50
max_{cost_cw}	1.50
max_{cost_px}	100.00
$pref_{cost\_ac}$	1.05
$pref_{cost\_cw}$	1.05
$pref_{cost\_px}$	10.00
max_{delay_ac}	120.00
max_{delay_cw}	120.00
$max_{tripTime\_px}$	120.00
$pref_{delay\_ac}$	0.00
$pref_{delay\_cw}$	0.00
$pref_{tripTime\_px}$	0.00

Anexo C

Experiências

Tabela C.1: Dados Reais da TAP Portugal - Tipo de informação disponível na base de dados.

Nome da Tabela	Descrição
Activities	Registo de actividades dos membros da tripulação.
Aircraft Models	Informações relativas aos diferentes modelos de avião: custos médios de Controlo de Tráfego Aéreo, manutenção, combustível e *handling*.
Aircrafts	Registo de aviões.
Airport Charges	As taxas aplicadas pelos diferentes aeroportos.
City Pairs	Latitude, longitude e distância entre dois aeroportos.
Crew Members	Grupo, posto e horas de voo para cada membro da tripulação.
Events	Eventos que causaram problemas no plano operacional.
Flights	Voos agendados.
Hotel Charges	Custos de hotel para membros da tripulação e passageiros.
Salaries	Informação sobre salários dos membros da tripulação.

Tabela C.2: Plano Operacional - Caracterização do plano operacional.

Característica	Descrição
Número de voos (*flight*)	7931
Capacidade total de lugares	1091974
Total de lugares vendidos	585744
Total de lugares disponíveis	506230
Membros de tripulação (*crew member*)	3028
Número de aviões (*aircraft*)	39 NB (19 A319, 17 A320, 3 A321) 16 WB (12 A330, 4 A340)

Tabela C.3: Problemas no Plano Operacional - Caracterização dos problemas que afectam o plano operacional.

Característica	**Descrição**
Voos afectados	49
Tipo de avião afectado	27 NB (14 A319, 10 A320, 3 A321) 4 WB (3 A330, 1 A340)
Tripulação afectada	286
Passageiros afectados	576 BC e 4184 YC
Total de minutos de atraso de voos	1752 minutos (média de 35,76 minutos por voo atrasado)
Custo total previsto da perspectiva *Aircraft*	93800 u.m. (média de 1914,29 u.m. por voo atrasado)
Custo total previsto da perspectiva *Crew Member*	98843 u.m. (média de 2017,20 u.m. por voo atrasado)
Causas (ver tabela A.1)	6 *airp*, 9 *atc*, 2 *comm*, 1 *crot*, 8 *hand*, 1 *induty*, 5 *maint*, 2 *meteo*, 1 *other*, 1 *rules*, 5 *sec*, 1 *sign*, 7 *rot*

Referências

[AT09] Adrian Agogino e Kagan Tumer. Improving air traffic management through agent suggestions. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pages 1271–1272, Richland, SC, 2009. International Foundation for Autonomous Agents and Multiagent Systems. URL: http://dl.acm.org/citation.cfm?id=1558109.1558247.

[Bal98] Tucker Balch. Behavioral diversity in learning robot teams. Technical report, 1998. URL: http://hdl.handle.net/1853/6637.

[BH05] Jeremy W. Baxter e Graham S. Horn. Controlling teams of uninhabited air vehicles. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, AAMAS '05, pages 27–33, New York, NY, USA, 2005. ACM. URL: http://doi.acm.org/10.1145/1082473.1082800, doi:10.1145/1082473.1082800.

[Bou96] Craig Boutilier. Learning conventions in multiagent stochastic domains using likelihood estimates. In *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*, pages 106–114. Morgan Kaufmann Publishers Inc., 1996.

[Cas13] Antonio J. M. Castro. *A Distributed Approach to Integrated and Dynamic Disruption Management in Airline Operations Control*. Phd thesis, Faculty of Engineering, University of Porto, Portugal, March 2013.

[CB03] Georgios Chalkiadakis e Craig Boutilier. Coordination in multiagent reinforcement learning: a bayesian approach. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, AAMAS '03, pages 709–716, New York, NY, USA, 2003. ACM. URL: http://doi.acm.org/10.1145/860575.860689, doi:10.1145/860575.860689.

[CG04] Jacob W. Crandall e Michael A. Goodrich. Multiagent learning during on-going human-machine interactions: The role of reputation, 2004.

[CLLR10] Jens Clausen, Allan Larsen, Jesper Larsen e Natalia J. Rezanova. Disruption management in the airline industry-concepts, models and methods. *Comput. Oper. Res.*, 37(5):809–821, 2010. doi:http://dx.doi.org/10.1016/j.cor.2009.03.027.

[CO11] Antonio J. M. Castro e Eugenio Oliveira. A new concept for disruption management in airline operations control. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 3(3):269–290, March 2011. doi:10.1243/09544100JAERO864.

[CRO12] Antonio J. M. Castro, Ana Paula Rocha e Eugenio Oliveira. Towards an autonomous and intelligent airline operations control. In *Proceedings of the 2012 15th IEEE Conference on Intelligent Transportation Systems (ITSC 2012)*, pages 1429–1434, Anchorage, Alaska, USA, September 16-19 2012.

[ESB10] Niklaus Eggenberg, Matteo Salani e Michel Bierlaire. Constraint-specific recovery network for solving airline recovery problems. *Comput. Oper. Res.*, 37(6):1014–1026, June 2010. doi:http://dx.doi.org/10.1016/j.cor.2009.08.006.

[Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, 1 edition, January 1989. URL: `http://www.worldcat.org/isbn/0201157675`.

[GR95] Claudia V. Goldman e Jeffrey S. Rosenschein. Mutually supervised learning in multiagent systems. In *Adaptation and Learning in Multi-Agent Systems*, pages 85–96. Springer-Verlag, 1995.

[JW03] Thomas Jansen e R.Paul Wiegand. Exploring the explorative advantage of the cooperative coevolutionary (1+1) ea. In Erick Cantú-Paz, JamesA. Foster, Kalyanmoy Deb, LawrenceDavid Davis, Rajkumar Roy, Una-May O'Reilly, Hans-Georg Beyer, Russell Standish, Graham Kendall, Stewart Wilson, Mark Harman, Joachim Wegener, Dipankar Dasgupta, MitchA. Potter, AlanC. Schultz, KathrynA. Dowsland, Natasha Jonoska e Julian Miller, editors, *Genetic and Evolutionary ComputationGECCO 2003*, volume 2723 of *Lecture Notes in Computer Science*, pages 310–321. Springer Berlin Heidelberg, 2003. URL: `http://dx.doi.org/10.1007/3-540-45105-6_37`.

[KLM96] L.P.a Kaelbling, M.L.a Littman e A.W.b Moore. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4:237–285, 1996. URL: `http://www.scopus.com/inward/record.url?eid=2-s2.0-0029679044&partnerID=40&md5=e29370fbf7ec7270107731432ecfd9e1`.

[NO04] Luís Nunes e Eugénio Oliveira. Learning from multiple sources. In *In AAMAS-2004 — Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems*, pages 1106–1113. IEEE Computer Society, 2004.

[NO05] Luís Nunes e Eugénio C. Oliveira. Advice-exchange between evolutionary algorithms and reinforcement learning agents: Experiments in the pursuit domain. In *Adaptive Agents and Multi-Agent Systems'05*, pages 185–204, 2005.

[Oli96] J.R. Oliver. On artificial agents for negotiation in electronic commerce. In *System Sciences, 1996., Proceedings of the Twenty-Ninth Hawaii International Conference on,*, volume 4, pages 337–346. IEEE, 1996.

[Pin12] M.L. Pinedo. *Scheduling: theory, algorithms, and systems*. Springer, 2012.

[PL05] Liviu Panait e Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11:387–434, 2005. URL: `http://dx.doi.org/10.1007/s10458-005-2631-2`, doi:10.1007/s10458-005-2631-2.

[PSJ+10] Jon D. Petersen, Gustaf Solveling, Ellis J. Johnson, Jonh-Paul Clarke e Sergey Shebalov. An optimization approach to airline integrated recovery. Technical report, The

Airline Group of the International Federation of Operational Research (AGIFORS), May 2010.

[PSW00] R. Parasuraman, T.B. Sheridan e C.D. Wickens. A model for types and levels of human interaction with automation. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 30(3):286–297, 2000.

[PW08] R. Parasuraman e C.D. Wickens. Humans: Still vital after all these years of automation. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(3):511–520, 2008.

[RN09] Stuart Russell e Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd edition, December 2009.

[Roc01] Ana Paula Rocha. *Metodologias de Negociação em Sistemas Multi-Agentes para Empresas Virtuais*. Phd thesis, Faculty of Engineering, University of Porto, December 2001.

[San99] Tuomas W Sandholm. Distributed rational decision making. *Multiagent systems: A modern approach to distributed artificial intelligence*, pages 201–258, 1999.

[SB98] Richard S. Sutton e Andrew G. Barto. *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning)*. The MIT Press, March 1998. URL: `http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0262193981`.

[SG99] Dicky Suryadi e Piotr J. Gmytrasiewicz. Learning models of other agents using influence diagrams. In *In Proceedings of the Seventh International Conference On User Modeling (UM-99*, pages 223–232, 1999.

[Sil13] José Silva. Case-based reasoning (cbr): Aprender com o passado no controlo de operações aéreas. January 2013.

[SS05] AlexanderA. Sherstov e Peter Stone. Function approximation via tile coding: Automating parameter choice. In Jean-Daniel Zucker e Lorenza Saitta, editors, *Abstraction, Reformulation and Approximation*, volume 3607 of *Lecture Notes in Computer Science*, pages 194–205. Springer Berlin Heidelberg, 2005. URL: `http://dx.doi.org/10.1007/1152786214`.

[SSK05] Peter Stone, Richard S. Sutton e Gregory Kuhlmann. Reinforcement learning for robocup-soccer keepaway. *Adaptive Behavior*, 13(3):165–188, 2005.

[SV78] T.B. Sheridan e W.L. Verplank. Human and computer control of undersea teleoperators. Technical report, DTIC Document, 1978.

[TCRO13] Francisca Teixeira, António J. M. Castro, Ana Paula Rocha e Eugénio Oliveira. Multi-agent learning in both cooperative and competitive environments. In *To appear in Proceedings of the XVI Portuguese Conference on Artificial Intelligence*, Angra do Heroísmo, Azores, Portugal, 9-12 September 2013.

[TF04] Keiki Takadama e Hironori Fujita. Q-learning and sarsa agents in bargaining game. In *in North American Association for Computational Social and Organizational Science (NAACSOS)*, 2004.

REFERÊNCIAS

[VNM07] J. Von Neumann e O. Morgenstern. *Theory of Games and Economic Behavior (Commemorative Edition)*. Princeton university press, 2007.

[WD92] C. J. C. H. Watkins e Peter Dayan. Q-learning. *Machine Learning*, 8:279–292, 1992.

[Woo09] M. Wooldridge. *An introduction to multiagent systems*. Wiley, 2009.